paperware 3

Universele terminal

Paperware 3 bevat aanvullende informatie over de Universele Terminal, onder gebruikmaking van de in Elektuur, september 1983 gepubliceerde VDU-kaart en de in Elektuur, november 1983 gepubliceerde CPU-kaart. Als CRT-controller (beeldsturing) wordt een IC van het type 6845 (Motorola) of 6545 (Rockwell/Synertek) gebruikt. Met dit IC kan op eenvoudige wijze een nabouwzekere video-interface worden gebouwd die tot véél in staat is.

De hoofdstukindeling is als volgt:

* I Beschrijving van de 6845/6545.
* II Programmeren van de 6845/6545.
* III Beschrijving en programmering van de ACIA 6551.
* IV Beschrijving van de diverse interfaces: V24/RS 232, parallel, Centronics.
* V Source-listing van de video-routines voor de Universele Terminal.



* I Beschrijving van de 6845/6545

Met het beeldsturings-IC (CRT-controller) type 6845/6545 kun je alle microcomputers op simpele wijze aansluiten op een video-monitor. Het IC is programmeerbaar; via software kunnen diverse beeldformaten worden gerealiseerd zonder ook maar één wijziging in de hardware van de video-interface. Daar komt nog bij dat het IC goedkoop is. Aangevuld met een paar TTL-IC's levert het IC alle beeldschrijfmiddelen, als onderdeel van de totale video-interface.
Figuur 1 toont de plattegrond en de interne opbouw (architektuur). Er zijn tellerschakelingen ten behoeve van de horizontale en de vertikale opbouw van het beeld(scherm). Verder een lineaire adresteller en een registergroep (bestaande uit een aantal programmeerbare registers). De tellerschakelingen zorgen eveneens voor de stuursignalen voor de video-monitor: HSYNC, VSYNC, DISPLAY ENABLE en de raster-adreslijnen RAØ . . . RA4. Met de 14 adreslijnen MAØ . . . MA13 kan beeldgeheugen tot maximaal 16 K worden geadresseerd.
De systeem-processor waarop de CRT-controller is aangesloten is in staat om de registergroep te lezen of er data in te schrijven. Dit levert de al genoemde mogelijkheid tot programmeren op. Men kan ook vastleggen hoeveel karakters een karakterrij omvat, uit hoeveel video-lijnen een karakterrij is opgebouwd, hoeveel karakterrijen, dus regels er op het scherm komen, of hoe breed de boven-, onder-, linker en rechter beeldrand moeten worden, of welke afmetingen de cursor (= "aanwijspunt") heeft. Of vastleggen dat de cursor knippert of juist niet. Zelfs de knipperfrekwentie van de cursor kan worden geprogrammeerd. Omdat er verschillende video-frekwenties, horizontaal en vertikaal, bestaan, afhankelijk van het land waarin men verblijft, zijn ook de tijdsafstanden tussen opeenvolgende horizontale, respektievelijk vertikale synchronisatie-pulsen programmeerbaar.

De aansluitingen van de 6845/6545

Pen 1 (massa) en pen 20 (+5 V)

Aansluitingen voor de voedingsspanning. De tolerantie in de voedingsspanning bedraagt ± 10%, dus het voedingsspanningsbereik loopt van 4,5 V tot 5,5 V.

Pen 2 ($\overline{RES}$)

Dit is de reset-ingang van de CRT-controller. Is aktief in de toestand "0". Tijdens het resetten wordt de inhoud van de display-startregisters R12 en R13 op de adresbus MAØ . . . . MA13 gezet. Verder worden alle interne tellers gereset en alle overige uitgangen "0" gemaakt. De inhoud van de interne registers blijft ongewijzigd.

Pen 3 (LPEN)

De lichtpen-ingang. Indien de lichtpen op een letter op het beeldscherm wordt geplaatst, dan geeft deze lichtpen een impuls af op tijdstippen dat de elektronenstraal van de monitor bezig is met het "schrijven" van die letter. De ingang LPEN reageert op een positieve lichtpen-impuls (dus "0"/"1"-overgang). Het bij de letter behorende adres wordt in het lichtpenregister (R16/R17) gekopiëerd. De processor is vervolgens in staat om het adres waarop de lichtpen staat gericht, te lezen.

Pennen 4 . . . 17 (MAØ . . . MA13)

Dit zijn de adreslijnen voor de adressering van video-RAM. Via deze adresbus worden opeenvolgende karakters in video-RAM aangewezen om op het scherm te worden weergegeven. Het momentane adres op deze adresbus bepaalt welk karakter (opgeslagen in de geadresseerde RAM-geheugenplaats) op dat moment op het scherm wordt weergegeven.

Pen 18 (DISPLAY ENABLE)

Het via de CRT-controller gerealiseerde beeld omvat een aktief gedeelte en een niet-aktief gedeelte. Indien de elektronenstraal bezig is met het schrijven van een beeldrand, dan spreken we van het niet-aktieve gedeelte. Dan is DISPLAY ENABLE "0". Is er daarentegen sprake van het schrijven van een karakterrij, dan betreft dit de aktieve fase en is DISPLAY ENABLE "1". Dit uitgangssignaal wordt gebruikt voor de "donkersturing" van het beeld en verder voor de sturing van de perifere logica van de CRT-controller.

Pen 19 (CURSOR)

Een uitgang, die "1" wordt zodra de CRT-controller die plaats op het beeldscherm adresseert, waar de cursor zich bevindt. De positie van de cursor op het scherm is vrij programmeerbaar. De inhoud van het cursor-register (R14/R15) is bepalend voor die positie. Het CURSOR-signaal wordt als ingang gebruikt voor de perifere logica ten behoeve van de cursor-sturing.

Pen 21 (CCLK)

Ingang voor het karakter-kloksignaal. De frekwentie van deze klok bepaalt de karakterfrekwentie op de video-monitor. Alle interne tellers gebruiken CCLK als tijdbasis. De puntfrekwentie (dot-frekwentie) is 8 x zo hoog als de karakterfrekwentie. Op de Elektuur-VDU-kaart is de karakterfrekwentie via deling herleid uit de puntfrekwentie. Hierdoor ontstaat een scherp en stabiel video-beeld.

Pen 22 (R/$\overline{W}$)

Deze ingang wordt gestuurd uit de R/$\overline{W}$-leiding van het processorsysteem. Indien data uit één van de interne registers moet worden gelezen, dan moet R/$\overline{W}$ "1" zijn. Een "0" op deze ingang maakt het voor de processor mogelijk om data in één van de interne registers te schrijven.

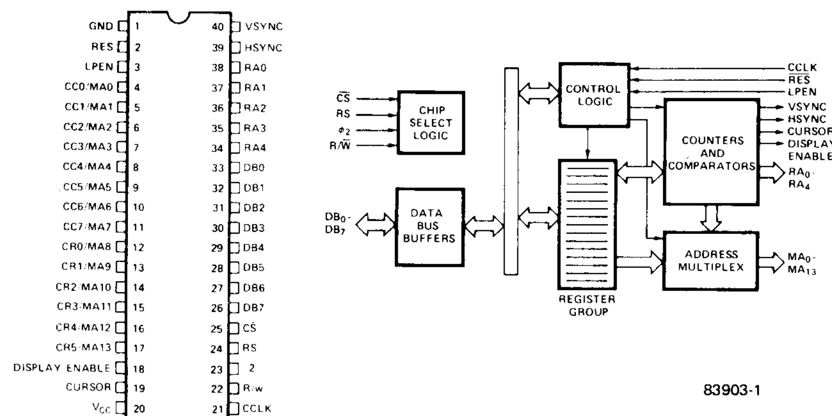
Pen 23 (Φ2 of E)

Dit is een ingang die wordt gestuurd uit de Φ2-systeemklok. Aangezien de Z80 een asynchrone busstruktuur bezit wordt deze pen op de VDU-kaart via een op de print aanwezige interface gestuurd. Alle data van of naar de CRT-controller wordt aan de hand van dit signaal gestuurd.

Pen 24 (RS)

Deze register-selektie-ingang is verbonden met adreslijn AØ van het processorsysteem. Via RS zijn alle interne registers toegankelijk. Indien RS "0" is wordt het adresregister van de CRT-controller geadresseerd. Het adresregister wijst het door de CPU te adresseren register aan. Als de processor dus een bepaald register binnen de 6845/6545 wil adresseren, dan moet eerst passende data in het adresregister worden geschreven en wel zodanig dat het adresregister op het gewenste register wijst. Zoals vermeld

1



83903-1

moet RS "0" zijn tijdens deze schrijfoperatie.
Nu is het gewenste register van de registergroep vastgelegd; in dit vastgelegde register kan de processor nu lezen of schrijven. Dan moet RS "1" zijn.

Pen 25 ($\overline{CS}$)

De Chip Select-ingang $\overline{CS}$ is via een adresdekoder met de adresbus van het processorsysteem verbonden. Is dit signaal "0", dan wordt de CRT-controller door de CPU geselekteerd.

Pennen 26 . . . 33 (DB7 . . . DBØ)

Deze acht leidingen zijn met de databus van het processorsysteem doorverbonden. Zolang de CRT-controller (via $\overline{CS}$) niet is geadresseerd zijn deze aansluitingen "tri state", dus feitelijk ontkoppeld.

Pennen 34 . . . 37 (RA4 . . . RAØ)

Vier raster-adreslijnen, uitgangen waarop de adressen voor de "raster scan" staan. De elektronenstraal op de videomonitor moet meerdere video-lijnen schrijven om één karakterrij op het scherm weer te geven. Het adres op deze vier lijnen geeft uitsluitsel over die video-lijnen die in het kader van de weergave van een karakterrij op een bepaald tijdstip wordt geschreven. Het aantal video-lijnen per karakterrij is vrij te kiezen, c.q. te programmeren. Dat geldt ook voor de afstand (in een aantal video-lijnen uitgedrukt) tussen twee opeenvolgende karakterrijen.

Pen 39 (HSYNC)

Het gaat hierbij om de horizontale synchronisatie-uitgang van de CRT-controller. De breedte (= tijdsduur) van de horizontale synchronisatiepuls en het tijdsverschil tussen opeenvolgende synchronisatiepulsen zijn vrij te programmeren.

Pen 40 (VSYNC)

Dat kan niet missen: de vertikale synchronisatie-uitgang van de CRT-controller. De breedte (= tijdsduur) van de vertikale synchronisatiepuls en het tijdsverschil tussen opeenvolgende vertikale synchronisatiepulsen zijn vrij te programmeren.

Figuur 1 toont u eveneens de interne opbouw van de CRT-controller. De chip-select-logica maakt het lezen en schrijven rond de registergroep mogelijk. De databusbuffers (ont)koppelen de interne 6845/6545-databus van/aan de externe databus. De tellers en de komparatoren realiseren, in samenwerking met de data in de diverse registers van de registergroep, de diverse video-uitgangssignalen die nodig zijn voor de opbouw van het uiteindelijke videosignaal. De adres-multiplexer zorgt voor de adressen van de op de videomonitor weergegeven karakters.

* II De programmering van de 6845/6545

De registergroep is van bijzonder belang voor de programmeur. De van de groep deel uitmakende registers bevatten alle informatie, die voor de opbouw van het video-beeld van belang zijn. Bijvoorbeeld: het aantal karakters per regel, het aantal karakterrijen (= regels) per beeld, het aantal door de elektronenstraal te schrijven video-lijnen voor de weergave van één regel. Indien er een lichtpen is aangesloten, dan kan de processor door het lezen van het lichtpenregister (dat deel uitmaakt van de registergroep) nagaan op welk beeldadres het karakter staat, waarmee de lichtpen "kontakt maakt". Dit soort informatie en nog veel meer andere informatie vindt u in de registers van de registergroep.
Figuur 2 laat zien hoe een (video-) beeld wordt opgebouwd. Horizontaal wordt het tijdsverloop in karakters geteld. Vertikaal telt "men" karakterrijen, dus regels. In het voorbeeld is sprake van een beeldformaat van 80 x 24 karakters. De karakters 0 . . . . 79 op de eerste rij van figuur 2 worden op het scherm weergegeven. De "karakters" 80 . . . 89 vertegenwoordigen de linker en rechter (donkere) beeldrand. De tijd, gedurende welke de CRT-controller van 0 tot 89 telt is vastgelegd op 64 μs. De puntfrekwentie is 8 x zo hoog als de karakterfrekwentie. Beide frekwenties zijn op de VDU-kaart gekoppeld via een gemeenschappelijke klokgenerator.
Een regel omvat meerdere, door de elektronenstraal te schrijven video-lijnen. Op de VDU-kaart is uitgegaan van 9 video-lijnen per regel, de donkere lijnen tussen twee opeenvolgende regels meegerekend. Een regel neemt derhalve 9 x 64 μs = 576 μs in beslag.
Vertikaal in figuur 2 worden regels, dus karakterrijen geteld. De rijen 0 . . . 23 betreffen het aktieve gedeelte van het beeld, de rijen 24 . . . 31 vormen de donkere boven- en onderrand van het beeld. De vertikale synchronisatiepuls VSYNC valt samen met het midden van het inaktieve regelgedeelte, de horizontale synchronisatiepuls HSYNC valt samen met het midden van het inaktieve karaktergedeelte.
Het beeldformaat en de positie van horizontale en vertikale synchronisatiepulsen kunnen vrij eenvoudig worden vastgelegd via het schrijven van data in registers van de registergroep. Daarom moet direkt na een systeemreset ($\overline{RES}$ = "Ø") door de processor data uit een opzoektabel (look up table) worden gelezen en in de registers van de registergroep worden gekopieerd. Over het programmeren van die registers gaan we het nu hebben.
De registergroep van de 6845/6545 omvat 21 registers, te weten RØ . . . . R19 en R31. Alle registers zijn hooguit acht bits breed. Alle registers worden gekozen via één enkele adreslijn.
Hoe gaat de registerkeuze in zijn werk? Eerst moet, via schrijven, het adresregister worden geladen met data die samenhangt met het gewenste register. In aansluiting daarop wordt de adreslijn van de CRT-controller (RS is aangesloten op AØ) "1" gemaakt en nu kan de processor via de databus data schrijven in het gewenste register. Voor de absolute adressen van het adresregister en van de registergroep wordt u verwezen naar de sourcelisting van de UT-software.

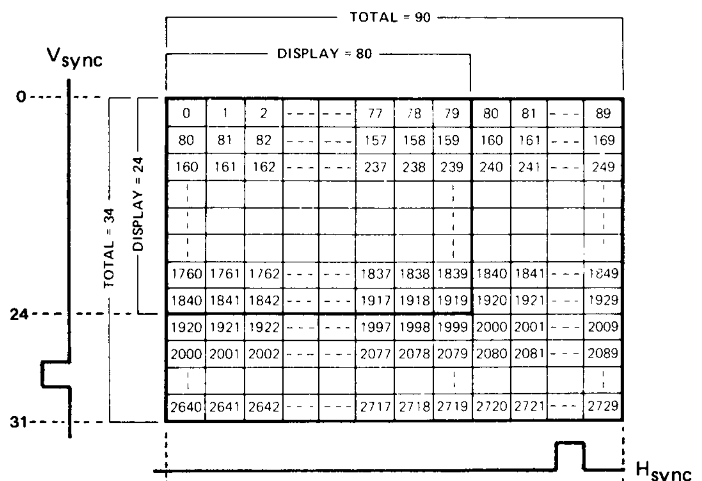
Register RØ (totaal horizontaal)

Dit register bevat in binaire vorm de optelsom van aktieve en niet-aktieve karakters binnen een regel, min één. Aktieve karakters zijn zichtbaar, niet-aktieve karakters stellen de linker en rechter beeldrand voor. De inhoud van RØ bepaalt de frekwentie van de horizontale synchronisatiepulsen.

Register R1 (horizontaal weergegeven)

Dit register bevat het aantal aktieve,

2



dus weergegeven karakters per regel. Een inhoud $5Ø duidt op 80 (aktieve) karakters per regel. De inhoud is nooit groter dan de inhoud van RØ.

Register R2
(positie horizontale syncpuls HSYNC)

De inhoud van dit register is bepalend voor de ligging van de horizontale synchronisatiepuls in een periode met niet-aktieve karakters, dus bepalend voor de breedte van de linker, respektievelijk rechter beeldrand. De inhoud van R2 ligt tussen die van R1 en die van RØ.

Register R3
(breedte van HSYNC & VSYNC)

De bits b3 . . . bØ van R3 bepalen hoeveel (maximaal 16) karakters de horizontale synchronisatiepuls breed is. De vertikale synchronisatiepuls neemt altijd 16 video-lijntijden in beslag. Met de SY 6545E van de firma Synertek is het mogelijk om via de bits b7 . . . b4 van R3 de breedte van de vertikale synchronisatiepuls VSYNC (maximaal 16 video-lijntijden) te programmeren.

Register R4 (totaal vertikaal)

Dit register bevat het aantal aktieve plus niet-aktieve karakterrijen (regels), min één. Een inhoud $1F van R4 duidt op een totaal-aantal van 32 rijen. De tijd die dan verstrijkt voordat alle 32 rijen, aktief of niet-aktief, van een beeld zijn geschreven, bedraagt 32 x 9 x 64 µs = 18,432 millisekonden. Aangezien de gebruikelijke tijd tussen twee opeenvolgende vertikale synchronisatiepulsen 20 millisekonden is, is een kleine korrektie van 20 minus 18,432 millisekonden noodzakelijk. Er ontbreekt 1,568 ms = 1568 µs. N.B. Met R5 vindt de aanvulling plaats in stappen van 64 µs.

Register R5 (aanpassing totaal vertikaal) (fijnafstemming)

Dit 5-bits register, dat niet kan worden gelezen, bepaalt het aantal lijntijden dat, opgeteld bij de inhoud van R4, zorg draagt voor een beeld-herhalingsfrekwentie van 50 Hz, dus een beeldtijd van 20 millisekonden.

Register R6 (vertikaal weergegeven)

Een 7-bits register dat het aantal aktieve karakterregels bevat. De inhoud van R5 is kleiner dan die van R4.

Register R7
(positie vertikale syncpuls VSYNC)

De inhoud van dit register bepaalt de positie van de vertikale synchronisatiepuls VSYNC binnen een aantal niet-aktieve karakterrijen (zie figuur 2). De positie wordt uitgedrukt in karaktertijden. Een en ander is van invloed op de hoogte van de bovenrand en van de onderrand van het beeld.

Register R8
(werkwijze van de CRT-controller)

Dit register omvat slechts twee bits. In figuur 3a is aangegeven hoe de bits de kompositie van het video-signaal beïnvloeden. Figuur 3c laat zien wat dat in de beeldpraktijk inhoudt. De R8 van de SY 6545E van Synertek biedt méér mogelijkheden. Hier is R8 acht bits breed. Zie hiervoor figuur 3b. De belangrijkste bits van dit uitgebreide R8-register zijn de "skew"-bits voor de cursor en voor het signaal DISPLAY ENABLE. Met deze twee bits kunnen de signalen CURSOR en DISPLAY ENABLE gedurende een karaktertijd worden vertraagd. Op de VDU-kaart is deze vertraging met flipflops, extern gerealiseerd. Voordeel: alle XX45 CRT-controllers kunnen op deze Elektuur-print worden ingezet!

Register R9
(aantal video-lijnen per regel)

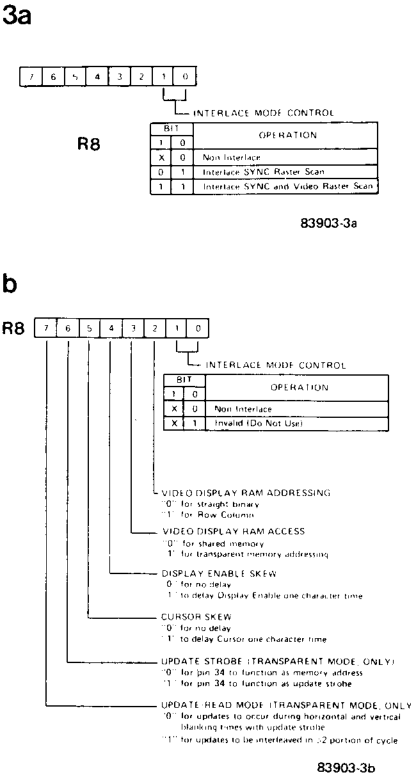
De inhoud van dit register bepaalt met hoeveel video-lijnen (minus 1) één karakterregel wordt beschreven door de elektronenstraal. Op de VDU-kaart is uitgegaan van 9 video-lijnen per regel, te weten acht aktieve lijnen en één "lege" video-lijn.
Let op! Indien grafische symbolen moeten worden weergegeven, moet worden uitgegaan van acht video-lijnen per grafisch teken. Dit omdat de grafische symbolen zijn opgebouwd volgens een 8 x 8-matrix. De "lege" video-lijn komt nu te vervallen, er ontstaat dan immers een aaneengesloten geheel van grafische symbolen.

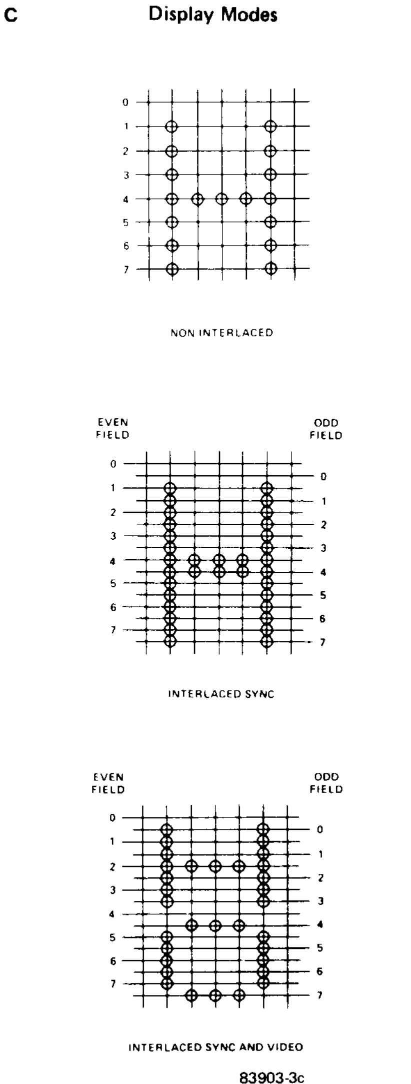
Registers R1Ø en R11
(cursor-informatie)

De rechter vijf bits van R10 bepalen, in welke video-lijn de cursor begint en de rechter vijf bits van R11 stellen vast in welke video-lijn de cursor eindigt (zie figuur 4b). Op deze wijze zijn dus de vertikale afmetingen van de cursor vastgelegd. De bits b6 en b5 van R10 geven antwoord op de vraag of de cursor knippert (met een bepaalde frekwentie) of niet knippert, en of de weergave van de cursor al dan niet moet plaatsvinden. Zie verder de waarheidstabel van figuur 4a.

Registers R12 (H) en R13 (L)
(Display-startadres)

Deze twee registers vormen tezamen een 14-bits adres, dat hoort bij de beginpositie van het eerste karakter ("home position", linksboven op het scherm). Op de VDU-kaart worden uitsluitend de bits bØ . . . b1Ø gebruikt omdat de video-RAM 2 K geheugenruimte omvat (ØØØ . . . 7FF).
Via manipulatie van de inhoud van R12 en R13 is het mogelijk om het beeld één regel omhoog of omlaag te laten verspringen, of om het beeld een met het aantal regels per beeld overeenkomend aantal regels omhoog of omlaag te laten verspringen. Men



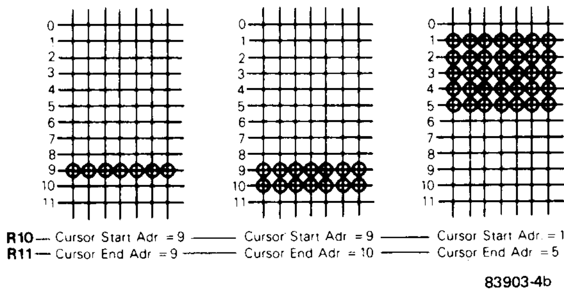


4a

R10	BIT		CURSOR MODE
	6	5	
	0	0	No Blinking
	0	1	No Cursor
	1	0	Blink at 16x field rate (fast)
	1	1	Blink at 32x field rate (slow)

83903-4a

b



83903-4b

spreekt van "scroll up" en "scroll down". In figuur 5 is de opbouw van de video-RAM geschetst. Dit geheugen kent een ringvormige opzet ("wrap around memory"). De sturing van alle karakters op het scherm geschiedt onder invloed van een hardware-adreswijzer (pointer), te weten de display-startadreswijzer, en drie software-adreswijzers: FLN, CLN en LLN. Alle vier genoemde adreswijzers zijn virtuele adreswijzers, m.a.w. hun inhoud komt niet overeen met het effektieve karakteradres in video-RAM. Het effektieve karakteradres bereken je als volgt:

effektieve karakteradres =
inhoud van CLN
(momentane cursor-regel)
plus
positienummer van de cursor
plus
absoluut beginadres van de video-RAM ($DØØØ)

Definities:

FLN = "First LiNe pointer". Staat altijd gericht op het begin van de eerste regel. Inhoud varieert van ØØØ. . . 7FF.

CLN = "Current LiNe pointer". Staat gericht op het begin van de regel waarin zich de cursor bevindt. De cursorpositie volgt uit de optelsom van inhoud CLN en de inhoud van COL, welke laatste het positienummer (kolomnummer) van de cursor bevat. De inhoud van CLN varieert van ØØØ . . . . 7FF.

LLN = "Last LiNe pointer". Staat altijd gericht op het begin van de onderste regel. Inhoud varieert van ØØØ . . . 7FF.

De display-startadreswijzer (in de CRT-controller) staat net als FLN gericht op het eerste karakter van de bovenste regel. Ook hier varieert de inhoud van ØØØ . . . 7FF.

De inhoud van de vier adreswijzers wijzigt zich als volgt:

1. Verhoog telkens met 1 tot maximaal $7FF. Bij overschrijding van de maximale inhoud begint het opnieuw bij nul.
2. Verlaag telkens met 1 tot minimaal nul. Begin vervolgens opnieuw bij $7FF.

Scroll:

Het gebruikelijke beeldformaat rond de Elektuur-VDU-kaart bedraagt 80 x 24. In vertikale richting kan de cursor bewegen tussen regel 1 en regel 24. Beweegt de cursor omlaag en voorbij de onderste regel, dan vindt een "scroll up" plaats. De video-software doet in dat geval de inhoud van het beeld een regel omhoog schuiven; de tot voor kort bovenste regel verdwijnt. Beweegt de cursor omhoog en voorbij de bovenste regel, dan is er sprake van een "scroll down". In dat geval wordt de beeldinhoud een regel omlaag verschoven; de tot voor kort onderste regel verdwijnt.

Er zijn twee soorten scroll:

1. Geheugen-scroll. De totale inhoud van de video-RAM wordt in het geheugen verplaatst; de verplaatsing komt overeen met het aantal karakters op een regel. Het display-startadres blijft ongewijzigd. In het geval van een 80 x 24-beeldformaat moet 80 x 24 = 1920 x data worden gekopieerd naar de nieuwe positie in video-RAM. Dat kost tijd en het houdt de processor teveel bezig. Deze metode wordt dan ook op de VDU-kaart niet gebruikt.
2. De adreswijzer-scroll. Bij deze gang van zaken wordt alleen de inhoud van het display-startadres (in de CRT-controller) gewijzigd. In het geval van een scroll up wordt de inhoud van het display-startadres met het aantal karakters per regel verhoogd, bij een scroll down met eenzelfde aantal verlaagd. Dit vergt slechts één schrijfoperatie voor de processor. Vandaar dat deze metode veel sneller verloopt.

Registers R14 (H) en R15 (L) (cursorspositie)

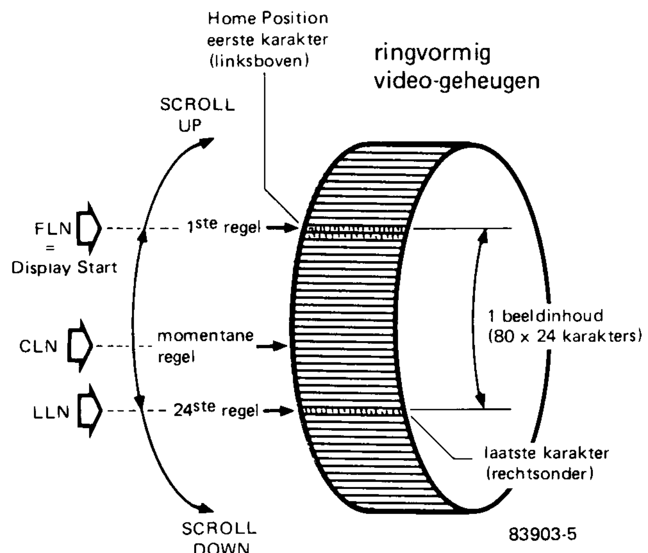
De inhoud is bepalend voor de positie van de cursor op het scherm. Zodra de data in R14/R15 gelijk is aan de door de adresbus MAØ . . . MA1Ø voorgestelde data wordt de uitgang CURSOR aktief. Verder worden automatisch de nivo's op de leidingen RAØ . . . RA4 vergeleken met de inhoud van de cursorregisters R1Ø/R11. Deze vergelijking levert de hoogte op van de cursor (uitgedrukt in een aantal videolijnen).

Registers R16 (H) en R17 (L) (lichtpenregisters)

Deze registers bevatten het adres van het karakter waarop de lichtpen staat gericht. De opname van dit adres gebeurt tijdens een nivo-overgang van "Ø" naar "1" op de ingang LPEN van de CRT-controller, gevolgd door een nivo-overgang van "1" naar "Ø" van het karakterkloksignaal.

De 6845/6545 bezit nog een paar andere registers, te weten R18, R19 en R31. Deze registers zullen hier niet worden besproken omdat ze niet dezelfde betekenis hebben voor al de diverse fabrikaten 6845/6545. Er wordt dan ook geen gebruik gemaakt van deze registers in de video-software.

5



83903-5

6

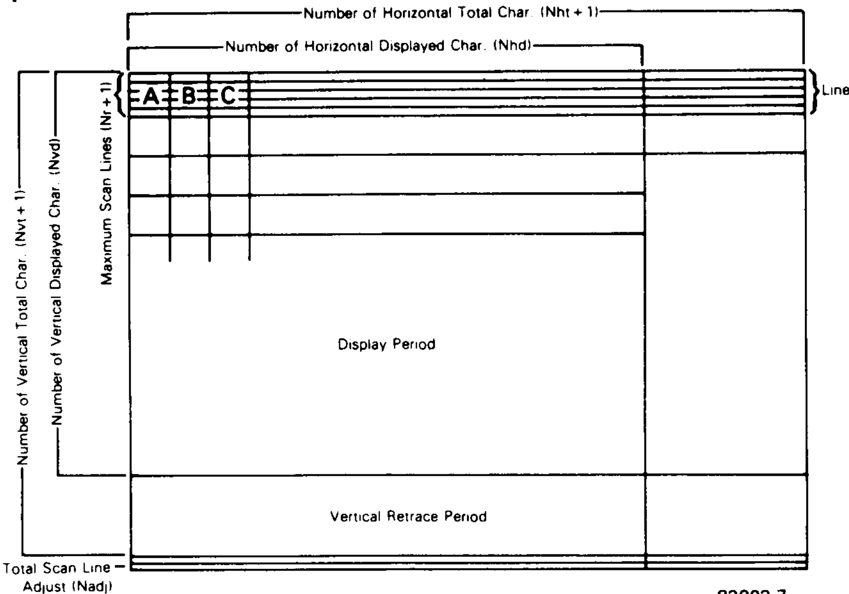
$\overline{CS}$	RS	Address Reg.					Reg. No.	Register Name	Stored Info.	RD	WR	Register Bit							
		4	3	2	1	0						7	6	5	4	3	2	1	0
1	–	–	–	–	–	–	–					▨	▨	▨	▨	▨	▨	▨	▨
0	0	–	–	–	–	–	–	Address Reg	Reg. No.		√	▨	▨	▨	A_4	A_3	A_2	A_1	A_0
0	0	–	–	–	–	–	–	Status Reg		√		U	L	V	▨	▨	▨	▨	▨
0	1	0	0	0	0	0	R0	Horiz. Total - 1	# Charac.		√	●	●	●	●	●	●	●	●
0	1	0	0	0	0	1	R1	Horiz. Displayed	# Charac.		√	●	●	●	●	●	●	●	●
0	1	0	0	0	1	0	R2	Horiz. Sync Position	# Charac.		√	●	●	●	●	●	●	●	●
0	1	0	0	0	1	1	R3	VSYNC, HSYNC Widths	# Scan Lines and # Char. Times		√	V_3	V_2	V_1	V_0	H_3	H_2	H_1	H_0
0	1	0	0	1	0	0	R4	Vert. Total - 1	# Charac. Row		√	▨	●	●	●	●	●	●	●
0	1	0	0	1	0	1	R5	Vert. Total Adjust	# Scan Lines		√	▨	▨	▨	●	●	●	●	●
0	1	0	0	1	1	0	R6	Vert. Displayed	# Charac. Rows		√	▨	●	●	●	●	●	●	●
0	1	0	0	1	1	1	R7	Vert. Sync Position	# Charac. Rows		√	▨	●	●	●	●	●	●	●
0	1	0	1	0	0	0	R8	Mode Control			√	●	●	●	●	●	●	●	●
0	1	0	1	0	0	1	R9	Scan Lines - 1	# Scan Lines		√	▨	▨	▨	●	●	●	●	●
0	1	0	1	0	1	0	R10	Cursor Start	Scan Line No.		√	▨	B_1	B_0	●	●	●	●	●
0	1	0	1	0	1	1	R11	Cursor End	Scan Line No.		√	▨	▨	▨	●	●	●	●	●
0	1	0	1	1	0	0	R12	Display Start Addr (H)			√	▨	▨	●	●	●	●	●	●
0	1	0	1	1	0	1	R13	Display Start Addr (L)			√	●	●	●	●	●	●	●	●
0	1	0	1	1	1	0	R14	Cursor Position (H)		√	√	▨	▨	●	●	●	●	●	●
0	1	0	1	1	1	1	R15	Cursor Position (L)		√	√	●	●	●	●	●	●	●	●
0	1	1	0	0	0	0	R16	Light Pen Reg (H)		√		▨	▨	●	●	●	●	●	●
0	1	1	0	0	0	1	R17	Light Pen Reg (L)		√		●	●	●	●	●	●	●	●
0	1	1	0	0	1	0	R18	Update Address Reg (H)			√	▨	▨	●	●	●	●	●	●
0	1	1	0	0	1	1	R19	Update Address Reg (L)			√	●	●	●	●	●	●	●	●
0	1	1	1	1	1	1	R31	Dummy Location				▨	▨	▨	▨	▨	▨	▨	▨

Notes: ● Designates binary bit

▨ Designates unused bit. Reading this bit is always "0", except for R31, which does not drive the data bus at all, and for CS = "1" which operates likewise.

83903-6

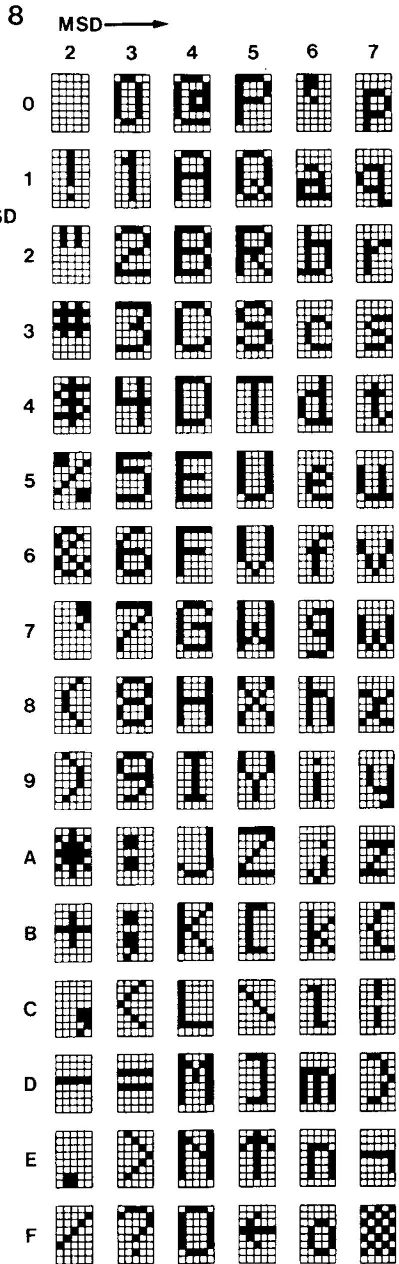
7



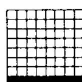
83903-7

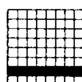
Een samenvattend overzicht van de registergroepen treft u aan in figuur 6. Figuur 7 verschaft een samenvattend overzicht van de beeldopbouw, met inbegrip van de diverse gebruikte begrippen.
Voor de omzetting van een karakter in het overeenkomende puntenpatroon zorgt de karaktergenerator. In een EPROM (2732) is het puntenpatroon vastgelegd van alle karakters en grafische symbolen. Aangezien Ohio Scientific het een en ander aan spelletjes-software te bieden heeft, zijn de grafische tekens kompatibel met Ohio.
De karaktergenerator bestaat uit twee helften, elk 2 K groot. De karakters beslaan het adresbereik ØØX . . . 7FX, de grafische tekens het adresbereik 8ØX . . . FFX (X = Ø . . . F). Figuur 8 toont de inhoud van de karaktergenerator voor de Amerikaanse versie en figuur 9 toont voor de volledigheid de wijzigingen ten behoeve van de Duitse versie (Umlauten en sz = ß).

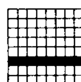
8

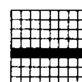


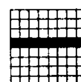
8

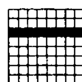

128 $8Ø

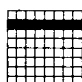

129 $81

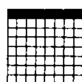

13Ø $82

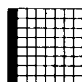

131 $83

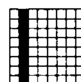

132 $84

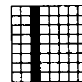

133 $85

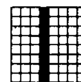

134 $86

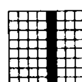

135 $87

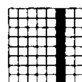

136 $88

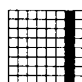

137 $89

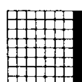

138 $8A

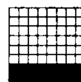

139 $8B

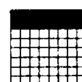

14Ø $8C


141 $8D

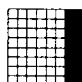

142 $8E

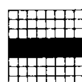

143 $8F

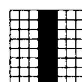

144 $9Ø

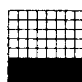

145 $91

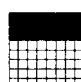

146 $92

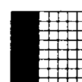

147 $93

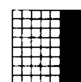

148 $94

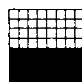

149 $95


15Ø $96

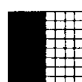

151 $97

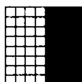

152 $98

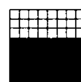

153 $99

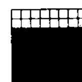

154 $9A


155 $9B


156 $9C

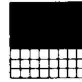

157 $9D

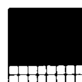

158 $9E


159 $9F

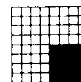

160 $AØ

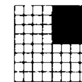

161 $A1

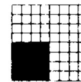

162 $A2

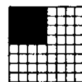

163 $A3


164 $A4


165 $A5

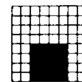

166 $A6

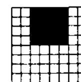

167 $A7

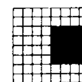

168 $A8

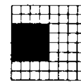

169 $A9

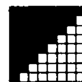

17Ø $AA

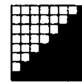

171 $AB

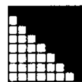

172 $AC


173 $AD

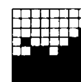

174 $AE


175 $AF

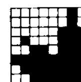

176 $BØ


177 $B1

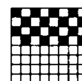

178 $B2

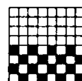

179 $B3

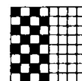

18Ø $B4

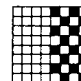

181 $B5

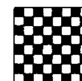

182 $B6

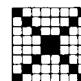

183 $B7

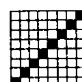

184 $B8

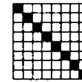

185 $B9

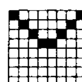

186 $BA

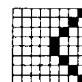

187 $BB

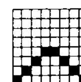

188 $BC

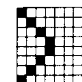

189 $BD

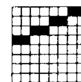

19Ø $BE

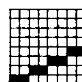

191 $BF

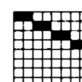

192 $CØ

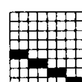

193 $C1

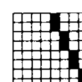

194 $C2

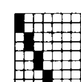

195 $C3

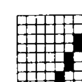

196 $C4

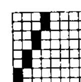

197 $C5

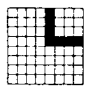

198 $C6

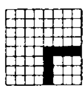

199 $C7

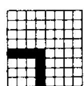

2ØØ $C8

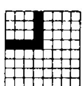

2Ø1 $C9

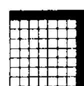

202 $CA

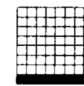

2Ø3 $CB

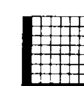

2Ø4 $CC

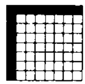

2Ø5 $CD

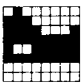

2Ø6 $CE

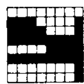

2Ø7 $CF

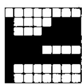

2Ø8 $DØ

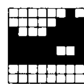

2Ø9 $D1


21Ø $D2

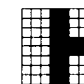

211 $D3

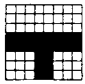

212 $D4

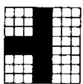

213 $D5

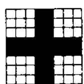

214 $D6

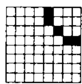

215 $D7

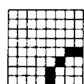

216 $D8

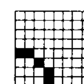

217 $D9

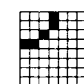

218 $DA

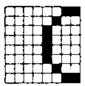

219 $DB

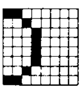

22Ø $DC

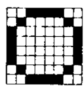

221 $DD

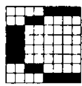

222 $DE

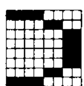

223 $DF

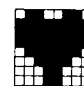

224 $EØ

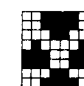

225 $E1

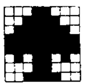

226 $E2

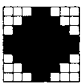

227 $E3

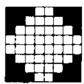

228 $E4

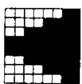

229 $E5

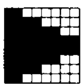

23Ø $E6

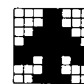

231 $E7

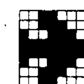

232 $E8

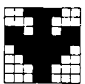

233 $E9

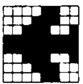

234 $EA

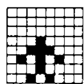

235 $EB

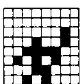

236 $EC

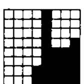

237 $ED

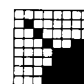

238 $EE

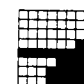

239 $EF

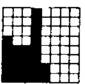

24Ø $FØ

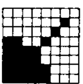

241 $F1

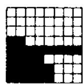

242 $F2

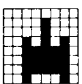

243 $F3

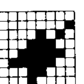

244 $F4

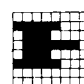

245 $F5

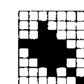

246 $F6

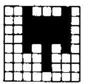

247 $F7

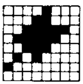

248 $F8

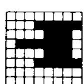

249 $F9

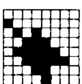

25Ø $FA


251 $FB


252 $FC


253 $FD

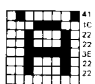

254 $FE


255 $FF

83903-8

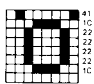
9

A



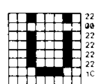
$ 5B

O



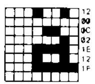
$ 5C

U



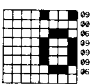
$ 5D

a



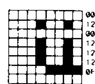
$ 7B

ö



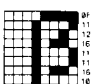
$ 7C

u



$ 7D

ß



$ 7E

83903 9

* III Beschrijving en programmering van de ACIA 6551

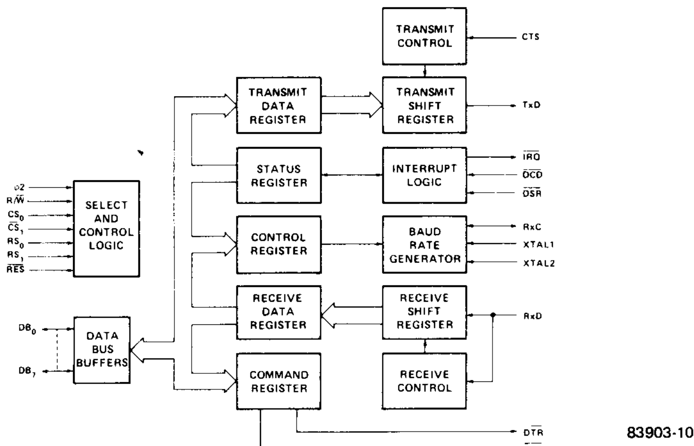
Het seriële dataverkeer tussen de computer en de Universele Terminal wordt geregeld door een ACIA, een asynchrone interface-adapter, van het type 6551. Dit IC herbergt een interne baudrate-generator, twee besturingsregisters, een statusregister, een dataverzendregister en een data-ontvangstregister. Extern zijn slechts een kristal en een nivo-aanpassing nodig. Figuur 10 toont het blokschema van de 6551. Alle handshake-leidingen, nodig voor de direkte aansluiting op een modem, zijn aanwezig.
Vijf interne registers van de 6551 worden geselekteerd door de signalen RSØ en RS1 (registerselektie), in sommige gevallen aangevuld met het signaal R/$\overline{W}$; figuur 11 geeft de details. De processor schrijft de te verzenden data in het data-verzendregister. De snelheid waarmee seriële data bit voor bit de ACIA in of uit gaat wordt bepaald door een programmeerbare baudrategenerator. Data die op het beeldscherm zichtbaar moet worden gemaakt ontvangt de Universele Terminal van de computer door het lezen van het data-ontvangstregister. De baudrate-generator stuurt de zender en de ontvanger binnen de ACIA.
Het is echter ook mogelijk om de ontvanger van de baudrate-generator te ontkoppelen en een externe baudrate-generator aan te sluiten. De zender en de ontvanger kunnen in dat geval met onderling verschillende transmissiesnelheden werken.
Voordat de tot de Universele Terminal behorende processor data in het dataverzendregister kan schrijven, of voordat-ie het data-ontvangstregister kan lezen, moet het statusregister worden gelezen. De bits van het statusregister geven uitsluitsel over de vraag of het data-verzendregister al leeg is, en of het data-ontvangstregister vol is. Andere bits van het statusregister stellen vast of er in de transmissie fouten zijn opgetreden of niet.
Het kommandoregister en het stuurregister zijn lees/schrijfregisters. Ze bepalen de werkwijze van de ACIA. Na een automatische of een manuele systeem-reset ($\overline{RES}$ = Ø) krijgen de beide registers via software een bepaalde inhoud toegekend; de werkwijze ligt daarmee vast.

De bits van het kommandoregister:

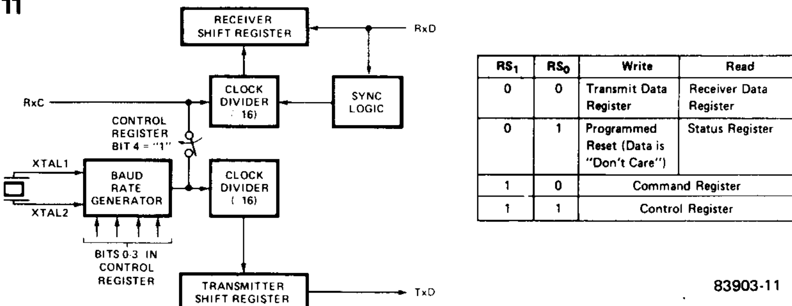
bØ:
- Ø = blokkeer een ontvanger-interrupt en alle andere interrupts.
- 1 = sta een ontvanger-interrupt en alle andere interrupts toe.

b1:
- Ø = sta een interrupt ten gevolge van een vol data-ontvangstregister toe.
- 1 = blokkeer een interrupt ten gevolge van een vol data-ontvangstregister.

10

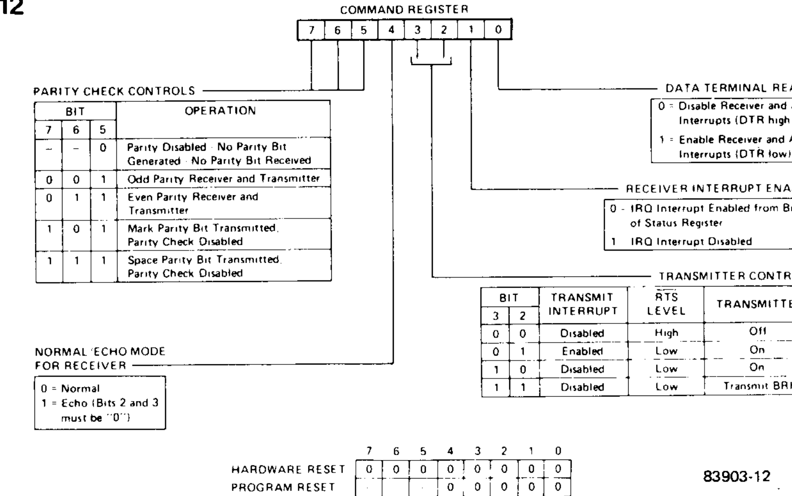


11



RS_1	RS_0	Write	Read
0	0	Transmit Data Register	Receiver Data Register
0	1	Programmed Reset (Data is "Don't Care")	Status Register
1	0	Command Register	
1	1	Control Register	

12

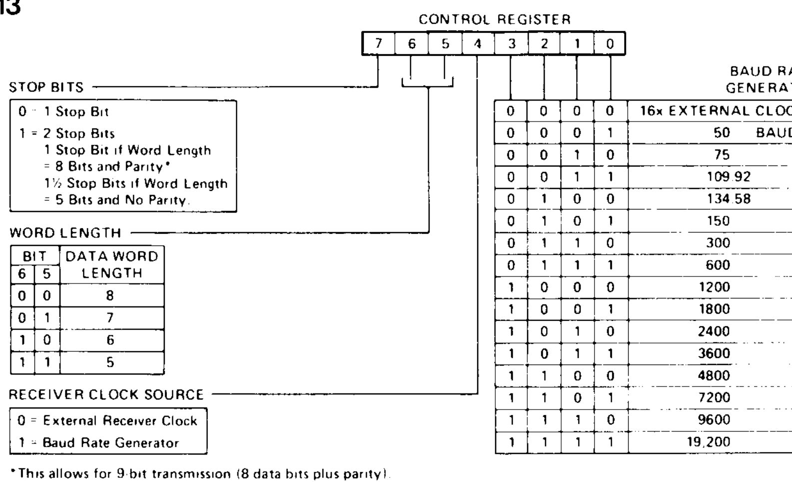


BIT 7	BIT 6	BIT 5	OPERATION
–	–	0	Parity Disabled · No Parity Bit Generated · No Parity Bit Received
0	0	1	Odd Parity Receiver and Transmitter
0	1	1	Even Parity Receiver and Transmitter
1	0	1	Mark Parity Bit Transmitted, Parity Check Disabled
1	1	1	Space Parity Bit Transmitted, Parity Check Disabled

BIT 3	BIT 2	TRANSMIT INTERRUPT	$\overline{RTS}$ LEVEL	TRANSMITTER
0	0	Disabled	High	Off
0	1	Enabled	Low	On
1	0	Disabled	Low	On
1	1	Disabled	Low	Transmit BRK

	7	6	5	4	3	2	1	0
HARDWARE RESET	0	0	0	0	0	0	0	0
PROGRAM RESET				0	0	0	0	0

13



BIT 6	BIT 5	DATA WORD LENGTH
0	0	8
0	1	7
1	0	6
1	1	5

3	2	1	0	BAUD RATE GENERATOR
0	0	0	0	16x EXTERNAL CLOCK
0	0	0	1	50 BAUD
0	0	1	0	75
0	0	1	1	109.92
0	1	0	0	134.58
0	1	0	1	150
0	1	1	0	300
0	1	1	1	600
1	0	0	0	1200
1	0	0	1	1800
1	0	1	0	2400
1	0	1	1	3600
1	1	0	0	4800
1	1	0	1	7200
1	1	1	0	9600
1	1	1	1	19,200

*This allows for 9-bit transmission (8 data bits plus parity).

	7	6	5	4	3	2	1	0
HARDWARE RESET	0	0	0	0	0	0	0	0
PROGRAM RESET	–	–	–	–	–	–	–	–

b3 b2:
Ø Ø = blokkeer een zender-interrupt. De $\overline{RTS}$-leiding is "1".
Ø 1 = sta een zender-interrupt toe. De $\overline{RTS}$-leiding is "Ø".
1 Ø = blokkeer een zender-interrupt, met $\overline{RTS}$ "Ø".
1 1 = blokkeer een zender-interrupt. met $\overline{RTS}$ "Ø". Verzend een BRK.

b4:
Ø = normale werkwijze
1 = echo-karakters. Omschakeling op half en full duplex. De bits b3 en b2 moeten "Ø" zijn.

b7 b6 b5:
X X Ø geen pariteitsbit verzonden of ontvangen.
Ø Ø 1 ontvang en verzend met oneven pariteit.
Ø 1 1 ontvang en verzend met even pariteit
1 Ø 1 verzend een mark-pariteitsbit; geen pariteitskontrole.
1 1 1 verzend een space-pariteitsbit; geen pariteitskontrole.

De diverse bits van het stuurregister (figuur 13):
b3 b2 b1 bØ:
Via deze vier bits wordt de baudrate van de ACIA geprogrammeerd. Indien alle vier bits nul zijn wordt er uitgegaan van een externe baudrate-generator, aangesloten op pen 5 (R x C). Er zijn 15 verschillende baudrates mogelijk van 50 tot en met 19200 baud.

b4:
Ø = ontvanger wordt met externe baudrate-generator geklokt, met pen 5 van de 6551 als klokingang.
1 = ontvanger en zender worden gestuurd vanuit de interne baudrate-generator, waarbij pen 5 van de 6551 de klokuitgang is. Omdat de interne baudrate-generator andere ACIA's kan sturen is het mogelijk om voor die andere ACIA's geen dure kristallen in te zetten.

b6 b5:
Ø Ø = serieel signaal met 8 databits.
Ø 1 = serieel signaal met 7 databits.
1 Ø = serieel signaal met 6 databits.
1 1 = serieel signaal met 5 databits (Baudot-transmissie).
Bit b7 bepaalt het aantal stopbits.

Figuur 14 toont de bits van het statusregister van de 6551. De belangrijkste bits zijn de ontvangstvlag b3 en de verzendvlag b4. Deze vlaggen vertellen of er een nieuw karakter mag worden verzonden of ontvangen. In de sourcelisting die van deze paperware deel uitmaakt vindt u typische verzend- en ontvangstroutines. De ACIA-verzendroutine heet ACIOUT, de ACIA-ontvangstroutine ACIIN. In figuur 15 is de aansluitplattegrond van de 6551 te zien, alsmede een ASCII-tabel.
Programmering van de ACIA:
De Universele Terminal van Elektuur bestaat uit de 6502-CPU-kaart, beschreven in Elektuur, november 1983, en de VDU-kaart, beschreven in Elektuur, september 1983. Beide kaarten zijn uitgevoerd in euroformaat en kunnen direkt op de Elektuurbus worden geplaatst. Opdat de Universele Terminal op alle computersystemen kan worden aangesloten, is op de CPU-kaart voorzien in aanvullende hardware ten behoeve van de eenvoudige programmering van het ACIA-stuurregister en van het ACIA-kommandoregister. In figuur 16 is te zien dat PL3 het ACIA-kommandoregister programmeert en dat PL4 het ACIA-stuurregister programmeert. De programmering bestaat uit het aanbrengen respektievelijk achterwege laten van draadbruggen.
Let op!
1 = draadbrug aangebracht;
Ø = kontakten open laten.
De al dan niet aangebrachte draadbruggen in PL3 en PL4 korresponderen met de afzonderlijke bits van het kommandoregister en van het stuurregister. Na een RESET leest de terminal het gekozen bitpatroon op PL3 en op PL4 en programmeert daarmee de werkwijze van de ACIA.
Hoe programmeer je het video-formaat?
Door in figuur 16 rond PL6 draadbruggen aan te brengen, respektievelijk niet aan te brengen is het mogelijk om het video-formaat vast te leggen, dus het aantal karakters per regel en het aantal regels per scherm. Er zijn acht video-formaten in EPROM vastgelegd; één daarvan is naar keuze van de gebruiker. De volgende formaten staan ter beschikking:

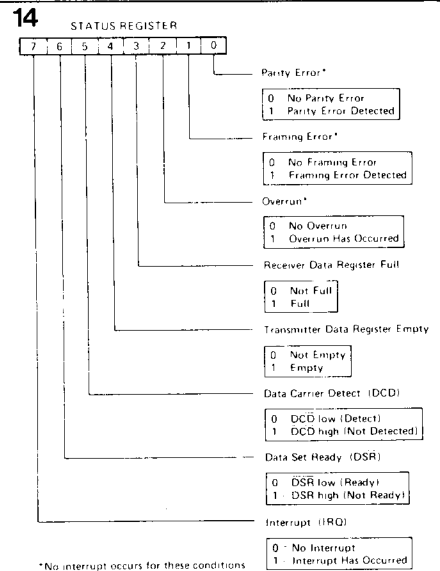
Ø Ø Ø 80 x 24
Ø Ø 1 80 x 25
Ø 1 Ø 64 x 16
Ø 1 1 64 x 24
1 Ø Ø 90 x 22
1 Ø 1 48 x 12
1 1 Ø 24 x 24
1 1 1 vrij te programmeren

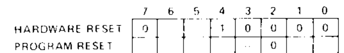
* IV Beschrijving van de diverse interfaces

a. RS 232/V24-interface

De seriële interface (= koppeling) van de Universele Terminal is in figuur 17 te zien. De totale seriële data-uitwisseling vindt plaats via PL7. Uit deze figuur wordt ook duidelijk hoe de 25-polige konnektor (mannetje oftewel male) op PL7 moet worden aangesloten, via een platte meeraderige verbinding (flat cable). Zolang er geen sprake is van de aansluiting op een modem moeten de pennen 4, 5, 6 en 8 op de konnektor onderling worden doorverbonden. Het Data-terminal-ready-signaal ligt aan pen 20

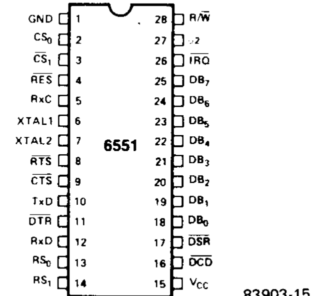
14





83903-14

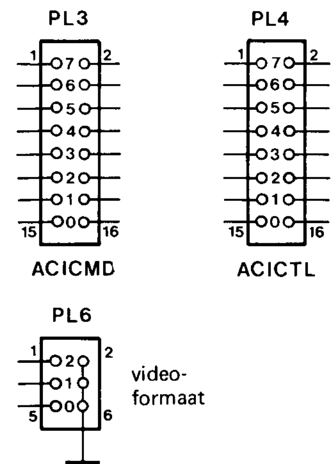
15



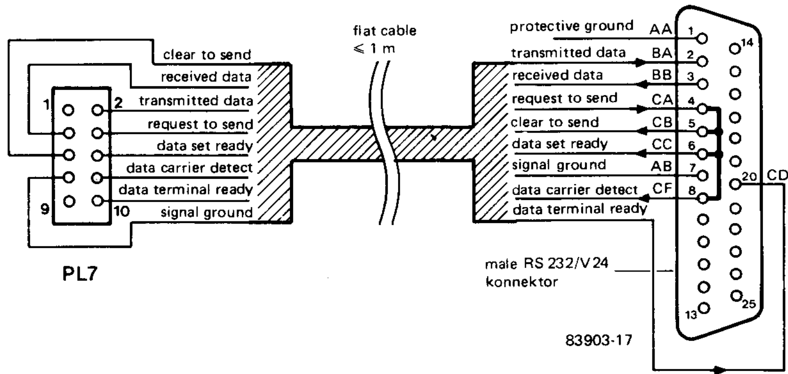
83903-15

MSD / LSD		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	0000	NUL	DLE	SP	0	@	P		p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENG	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*		J	Z	j	z
B	1011	VT	ESC	+	,	K	[	k	
C	1100	FF	FS	'	<	L	\	l	|
D	1101	CR	GS	–	=	M	]	m	
E	1110	SO	RS	•	>	N	↑	n	~
F	1111	SI	VS	/	?	□	←	o	DEL

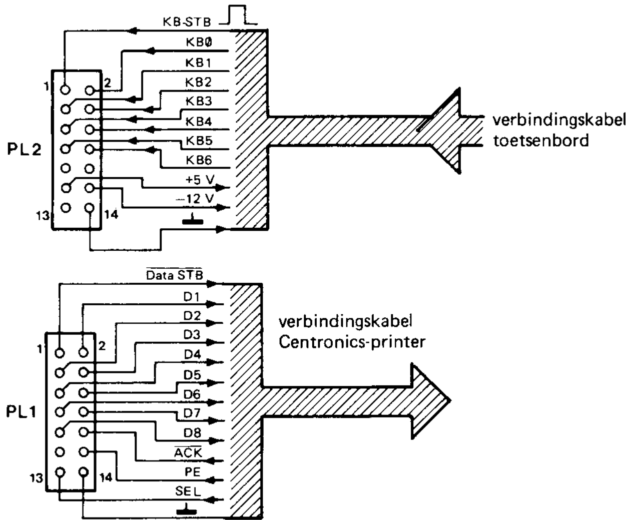
16



17



18

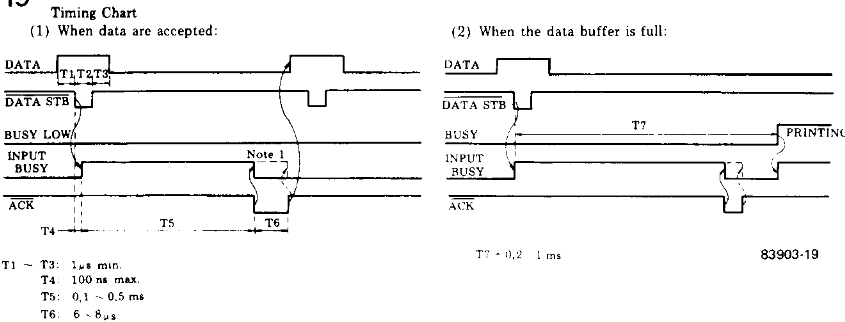


Pin No.	Signal Name	Pin No.	Signal Name
1	DATA STB	19	TWISTED PAIR GND (PIN 1)
2	DATA 1	20	TWISTED PAIR GND (PIN 2)
3	DATA 2	21	TWISTED PAIR GND (PIN 3)
4	DATA 3	22	TWISTED PAIR GND (PIN 4)
5	DATA 4	23	TWISTED PAIR GND (PIN 5)
6	DATA 5	24	TWISTED PAIR GND (PIN 6)
7	DATA 6	25	TWISTED PAIR GND (PIN 7)
8	DATA 7	26	TWISTED PAIR GND (PIN 8)
9	DATA 8	27	TWISTED PAIR GND (PIN 9)
10	ACK	28	TWISTED PAIR GND (PIN 10)
11	INPUT BUSY	29	TWISTED PAIR GND (PIN 11)
12	PE	30	TWISTED PAIR GND (PIN 31)
13	SELECT	31	INPUT-PRIME
14	0 V	32	FAULT
15	NC	33	0 V
16	0 V	34	NC
17	CHASSIS GND	35	NC
18	+5 V DC	36	INPUT-BUSY

19 30 36
1 18

83903-18

19



en dient voor de melding van de terminal aan de computer, dat hij, d.w.z. de terminal, niet bereid is om op dat moment data te ontvangen. Dat verandert zodra pen 20 een +12 V-nivo aanneemt (negatieve logica).

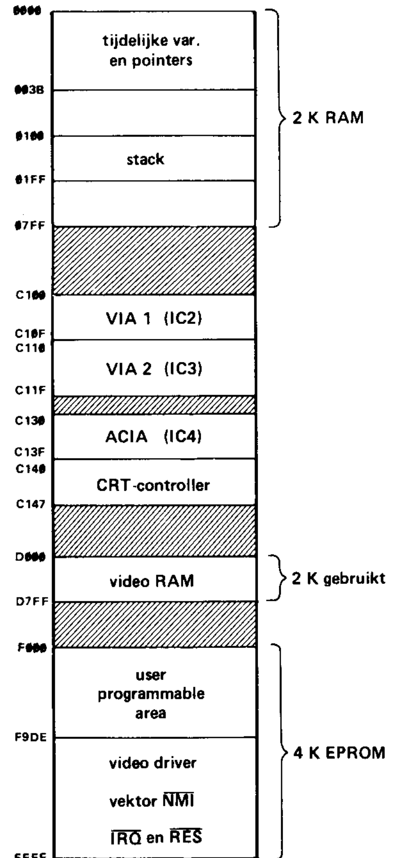
b. parallel-interface

Konnektor PL2 in figuur 18 is bedoeld voor de aansluiting van een parallel toetsenbord, met behulp van meeraderig "flat cable". De handshakeleiding CA1 van IC2 is doorverbonden met de keyboard-strobe (KB-STB). In het interrupt-vlagregister van de 6522-VIA wordt er een vlag "1" zodra een positieve KB-STB-puls binnenkomt. De toetsenbord-data KB0 . . . . KB6 wordt in het ingangs-tussenregister van de VIA opgeslagen. Dat heeft het voordeel dat de (ASCII-) waarde van een ingedrukte toets behouden blijft na het loslaten van de toets. Aan PL2 zijn de spanningen +5 V en −12 V toegevoerd. Het is dan ook mogelijk om uw toetsenbord uit de Universele Terminal te voeden.

c. Centronics-interface (printer)

Een printer, uitgerust met een Centronics-interface, kan via PL1 (zie figuur 18) direkt op de Universele Terminal worden aangesloten. Een garantie voor het probleemloos funktioneren van het dataverkeer biedt het aan massa leggen van de zogenaamde retourleidingen ("twisted ground"). Figuur 19 toont het tijdsverloop van de diverse handshakesignalen op de Centronics-stuurbus; deze signalen staan onder supervisie van de software van de Universele Terminal.
Figuur 20 geeft de geheugenindeling ("memory map") van de Universele Terminal weer.

20



```
F9DE            ORG    $F9DE


                SOURCE LISTING OF A VT-52 COMPATIBLE TERMINAL

                WRITTEN BY A. NACHTMANN


                COPYRIGHT C 1983 ELEKTOR/ELEKTUUR


                PROCESSOR:       6502 OR 65C02
                CRT CONTROLLER: 6845 MOTOROLA, ROCKWELL, SYNERTEC


                *VT-52 COMMANDS*

                OTHER COMMANDS SEE COMMAND LOOKUP TABLES


                CARRIAGE RETURN
                LINE FEED
                ESC H CURSOR HOME
                ESC A CURSOR UP
                ESC B CURSOR DOWN
                ESC C CURSOR RIGHT
                ESC D CURSOR LEFT
                ESC K ERASE TO END OF LINE
                ESC J ERASE TO END OF SCREEN

                *POINTERS AND TEMPS*

F9DE            RAMPTR *     $0000  RAMPOINTER
F9DE            AHOLD  *     $0002  SAVE THE CHARACTER HERE
F9DE            FLN    *     $0004  FIRST LINE POINTER
F9DE            CLN    *     FLN    +02 CURRENT LINE POINTER
F9DE            LLN    *     CLN    +02 LAST LINE POINTER
F9DE            SCRPTR *     LLN    +02 SLAVE SCREEN POINTER
F9DE            CURSOR *     SCRPTR +02 CURRENT CURSOR
F9DE            INLINE *     CURSOR +02 IN LINE WITH WINDOW
F9DE            COL    *     INLINE +01 CURRENT COLUMN
F9DE            ESCFLG *     COL    +01 ESC FLAG
F9DE            TEMCOL *     ESCFLG +01 SLAVE COLUMN
F9DE            RAMBEG *     TEMCOL +01 THE REFRESH RAM STARTS HERE
F9DE            CHAPLN *     RAMBEG +02 CHARACTER/LINE
F9DE            LPSCR  *     CHAPLN +01 LINES/SCREEN
F9DE            FORMAT *     LPSCR  +01 SCREEN FORMAT
F9DE            INDEX  *     FORMAT +01 SOFTWARE STACK FOR CHAR. INPUT
F9DE            BUFFER *     INDEX  +01
F9DE            TABLE  *     BUFFER +08 CRT FORMAT TABLE
F9DE            JMPVEC *     TABLE  +12 COMMAND ADDRESS POINTER
F9DE            CENFLG *     JMPVEC +02 CENTRONICS FLAG
F9DE            DUPLEX *     CENFLG +01 0=FULL;  1=HALF DUPLEX
F9DE            AUTOLF *     DUPLEX +01 0=NO; 1=AUTO CRLF TO CENTRONICS
F9DE            NMIVEC *     AUTOLF +01 (NMI VECTOR)
```

```
F9DE          IRQVEC *      NMIVEC +02 (IRQ VECTOR)

              *SPECIAL CHARACTERS*

F9DE          SPACE  *      $0020
F9DE          ESC    *      $001B

              *JUNIOR'S SERIAL I/0*

              *VIA 1 ADDRESSES*

F9DE          VAPBD  *      $C100  PORT B DATA
F9DE          VAPAD  *      VAPBD  +01 PORT A DATA
F9DE          VAPBDD *      VAPBD  +02 PORT B DATA DIRECTION
F9DE          VAPADD *      VAPBD  +03 PORT A DATA DIRECTION
F9DE          VATACL *      VAPBD  +04 T1, LATCH-LOW, COUNTER-LOW
F9DE          VATACH *      VAPBD  +05 T1, COUNTER-HIGH
F9DE          VATALL *      VAPBD  +06 T1, LATCH-LOW
F9DE          VATALH *      VAPBD  +07 T1, LATCH-HIGH
F9DE          VATBCL *      VAPBD  +08 T2, LATCH-LOW,COUNTER-LOW
F9DE          VATBCH *      VAPBD  +09 T2, COUNTER-HIGH
F9DE          VASR   *      VAPBD  +0A SHIFT REGISTER
F9DE          VAACR  *      VAPBD  +0B AUXILIARY CONTROL REGISTER
F9DE          VAPCR  *      VAPBD  +0C PERIPHERAL CONTROL REGISTER
F9DE          VAIFR  *      VAPBD  +0D INTERRUPT FLAG REGISTER
F9DE          VAIER  *      VAPBD  +0E INTERRUPT ENABLE REGISTER
F9DE          VAPADN *      VAPBD  +0F PORT A DATA, NO HANDSHAKE

              *VIA 2 ADDRESSES*

F9DE          VBPBD  *      $C110  PORT B DATA
F9DE          VBPAD  *      VBPBD  +01 PORT A DATA
F9DE          VBPBDD *      VBPBD  +02 PORT B DATA DIRECTION
F9DE          VBPADD *      VBPBD  +03 PORT A DATA DIRECTION
F9DE          VBTACL *      VBPBD  +04 T1, LATCH-LOW, COUNTER-LOW
F9DE          VBTACH *      VBPBD  +05 T1, COUNTER-HIGH
F9DE          VBTALL *      VBPBD  +06 T1, LATCH-LOW
F9DE          VBTALH *      VBPBD  +07 T1, LATCH-HIGH
F9DE          VBTBCL *      VBPBD  +08 T2, LATCH-LOW, COUNTER-LOW
F9DE          VBTBCH *      VBPBD  +09 T2, COUNTER-HIGH
F9DE          VBSR   *      VBPBD  +0A SHIFT REGISTER
F9DE          VBACR  *      VBPBD  +0B AUXILIARY CONTROL REGISTER
F9DE          VBPCR  *      VBPBD  +0C PERIPHERAL CONTROL REGISTER
F9DE          VBIFR  *      VBPBD  +0D INTERRUPT FLAG REGISTER
F9DE          VBIER  *      VBPBD  +0E INTERRUPT ENABLE REGISTER
F9DE          VBPADN *      VBPBD  +0F PORT A DATA ,NO HANDSHAKE

              *ACIA ADDRESSES*

F9DE          RECREG *      $C130  RECEIVER REGISTER
F9DE          TRAREG *      RECREG TRANSMITTER REGISTER
```

```
F9DE          ACIASR *     RECREG +01 STATUS REGISTER
F9DE          ACICMD *     RECREG +02 COMMAND REGISTER
F9DE          ACICTL *     RECREG +03 CONTROL REGISTER

              *CRT ADDRESSES*

F9DE          AR     *     $C140  ADDRESS REGISTER OF THE CRT
F9DE          RFILE  *     AR     +01 REGISTER FILE OF CRT

              ****************************
              * VT 52 COMPATIBLE TERMINAL *
              ****************************

              *INTERNAL 6845 REGISTER FILE*

F9DE          HORTOT *     $0000  HORIZONTAL TOTAL-1
F9DE          HORDIS *     $0001  HORIZONTAL DISPLAYED
F9DE          HSYPOS *     $0002  HORIZONTAL SYNC POSITION
F9DE          VHSYWI *     $0003  VERTICAL, HORIZONTAL SYNC WIDTH
F9DE          VERTOT *     $0004  VERTICAL TOTAL-1
F9DE          VTOTAJ *     $0005  VERTICAL TOTAL ADJUST
F9DE          VERDIS *     $0006  VERTICAL DISPLAYED
F9DE          VSYPOS *     $0007  VERTICAL SYNC POSITION
F9DE          MODE   *     $0008  INTERLACE MODE REGISTER
F9DE          SCANLN *     $0009  CHARACTER SCAN LINES-1
F9DE          CURSTA *     $000A  CURSOR START
F9DE          CUREND *     $000B  CURSOR END
F9DE          DSPSTH *     $000C  DISPLAY START HIGH
F9DE          DSPSTL *     $000D  DISPLAY START LOW
F9DE          CURPOH *     $000E  CURSOR POSITION HIGH
F9DE          CURPOL *     $000F  CURSOR POSITION LOW
F9DE          LIPENH *     $0010  LIGHT PEN HIGH
F9DE          LIPENL *     $0011  LIGHT PEN LOW
F9DE          UPDATH *     $0012  UPDATE HIGH (ROCKWELL, SYNERTEK)
F9DE          UPDATL *     $0013  UPDATE LOW
F9DE          DUMMY  *     $001F  DUMMY REGISTER

              ***VIDEO HANDLER***

F9DE 48       VIDEO  PHA          SAVE THE REGISTERS
F9DF 8A              TXA
F9E0 48              PHA
F9E1 98              TYA
F9E2 48              PHA
F9E3 A5 02           LDA   AHOLD  GET THE CHARACTER
F9E5 C9 1B           CMPIM ESC    IS IT THE ESC CHAR.?
F9E7 D0 15           BNE   CHECK
F9E9 85 10           STA   ESCFLG SET THE ESC FLAG
```

```
F9EB 68          VIDEND PLA          RESTORE THE REGISTERS
F9EC A8                 TAY
F9ED 68                 PLA
F9EE AA                 TAX
F9EF 68                 PLA
F9F0 EA                 NOP          $2C
F9F1 EA                 NOP          $80 BIT SPAD CHECK FOR BREAK
                                         DURING OUTPUT
F9F2 EA                 NOP          $FA OR $1A
F9F3 EA                 NOP          $10 BPL BRKTST
F9F4 EA                 NOP          $01
F9F5 60                 RTS
F9F6 EA          BRKTST NOP          $2C BIT SPAD WAIT TILL KEY
                                         IS RELEASED
F9F7 EA                 NOP          $80
F9F8 EA                 NOP          $FA OR $1A
F9F9 EA                 NOP          $10 BPL BRKTST
F9FA EA                 NOP          $FB
F9FB EA                 NOP          $6C JMP ($FA7C) GOTO BREAK HAND
F9FC EA                 NOP          $7C
F9FD EA                 NOP          $FA OR $1A
F9FE 20 E7 FC    CHECK  JSR   COMCOM COMPUTE THE COMMAND INDEX
FA01 B0 0E              BCS   VALVEC THERE WAS NO COMMAND
FA03 BD 4F FD           LDAX  COMADR SET THE COMMAND VECTOR
FA06 85 32              STA   JMPVEC
FA08 E8                 INX
FA09 BD 4F FD           LDAX  COMADR
FA0C 85 33              STA   JMPVEC +01
FA0E 6C 32 00           JMI   JMPVEC
FA11 4C 7F FA    VALVEC JMP   VALID
                 ***EXECUTE A VIDEO COMMAND***
                 *CARRIAGE RETURN*
FA14 A2 00       RETURN LDXIM $00
FA16 86 0F              STX   COL    COL=0
FA18 20 EE FA           JSR   ADJUST ADJUST THE CURSOR
FA1B 4C EB F9           JMP   VIDEND
                 *LINE FEED*
FA1E 20 52 FC    FEED   JSR   CURDN
FA21 4C EB F9           JMP   VIDEND
                 *CLEAR SCREEN & HOME CURSOR*
FA24 20 97 FC    CLRHOM JSR   HOME
FA27 20 D9 FB           JSR   ERTEOS
FA2A 4C EB F9           JMP   VIDEND
                 *CURSOR LEFT*
```

```
FA2D 20 8D FC   LEFT    JSR    CURLFT
FA30 4C EB F9           JMP    VIDEND
                *HOME CURSOR*
FA33 20 97 FC   HOCU    JSR    HOME
FA36 4C EB F9           JMP    VIDEND
                *CURSOR UP*
FA39 20 71 FC   UP      JSR    CURUP
FA3C 4C EB F9           JMP    VIDEND
                *CURSOR DOWN*
FA3F 20 52 FC   DOWN    JSR    CURDN
FA42 4C EB F9           JMP    VIDEND
                *CURSOR RIGHT*
FA45 20 92 FC   RIGHT   JSR    CURRGT
FA48 4C EB F9           JMP    VIDEND
                *ERASE TO END OF LINE*
FA4B 20 AE FB   ERLNX   JSR    ERTEOL
FA4E 4C EB F9           JMP    VIDEND
                *ERASE TO END OF SCREEN*
FA51 20 D9 FB   ERSCRX  JSR    ERTEOS
FA54 4C EB F9           JMP    VIDEND
                *DELETE THE CURRENT LINE*
FA57 A2 00      DELLIN  LDXIM  $00
FA59 86 0F              STX    COL
FA5B 20 35 FB           JSR    ADJCUR MOVE CURSOR TO COL 0
FA5E 20 AE FB           JSR    ERTEOL
FA61 4C EB F9           JMP    VIDEND
                *TOGGLE THE CENTRONICS FLAG*
FA64 A5 34      TOGGLE  LDA    CENFLG
FA66 49 01              EORIM  $01
FA68 85 34              STA    CENFLG
FA6A 4C EB F9           JMP    VIDEND
                *TOGGLE THE AUTO CRLF FLAG*
```

```
FA6D A5 36       TOGLF  LDA   AUTOLF
FA6F 49 01              EORIM $01
FA71 85 36              STA   AUTOLF
FA73 4C EB F9           JMP   VIDEND

                 *TOGGLE THE DUPLEX FLAG*

FA76 A5 35       TOGDUP LDA   DUPLEX
FA78 49 01              EORIM $01
FA7A 85 35              STA   DUPLEX
FA7C 4C EB F9           JMP   VIDEND



                 *FILTER < SPACE*

FA7F A5 02       VALID  LDA   AHOLD
FA81 C9 20              CMPIM '
FA83 B0 03              BCS   TOSCR
FA85 4C EB F9           JMP   VIDEND

                 *TRANSFER TO SCREEN*

FA88 20 29 FC    TOSCR  JSR   TVPUT
FA8B 4C EB F9           JMP   VIDEND



                 ***SUBROUTINES***


                 *FIRST LINE UP*

FA8E 38          FLNUP  SEC
FA8F A5 04              LDA   FLN
FA91 E5 14              SBC   CHAPLN FLN=FLN-CHAPLN
FA93 85 04              STA   FLN
FA95 A5 05              LDA   FLN    +01
FA97 E9 00              SBCIM $00
FA99 29 07              ANDIM $07    MAX IS $7FF
FA9B 85 05              STA   FLN    +01
FA9D 60                 RTS

                 *CURRENT LINE UP*

FA9E 38          CLNUP  SEC
FA9F A5 06              LDA   CLN
FAA1 E5 14              SBC   CHAPLN
FAA3 85 06              STA   CLN
FAA5 A5 07              LDA   CLN    +01
FAA7 E9 00              SBCIM $00
FAA9 29 07              ANDIM $07    MAX IS $7FF
FAAB 85 07              STA   CLN    +01
FAAD 60                 RTS

                 *LAST LINE UP*
```

```
FAAE 38       LLNUP   SEC
FAAF A5 08            LDA    LLN
FAB1 E5 14            SBC    CHAPLN
FAB3 85 08            STA    LLN
FAB5 A5 09            LDA    LLN     +01
FAB7 E9 00            SBCIM  $00
FAB9 29 07            ANDIM  $07     MAX IS $7FF
FABB 85 09            STA    LLN     +01
FABD 60               RTS

              *FIRST LINE DOWN*

FABE 18       FLNDN   CLC
FABF A5 04            LDA    FLN
FAC1 65 14            ADC    CHAPLN
FAC3 85 04            STA    FLN     FLN=FLN+CHAPLN
FAC5 A5 05            LDA    FLN     +01
FAC7 69 00            ADCIM  $00
FAC9 29 07            ANDIM  $07     MAX IS $7FF
FACB 85 05            STA    FLN     +01
FACD 60               RTS

              *CURRENT LINE DOWN*

FACE 18       CLNDN   CLC
FACF A5 06            LDA    CLN
FAD1 65 14            ADC    CHAPLN
FAD3 85 06            STA    CLN     CLN=CLN+CHAPLN
FAD5 A5 07            LDA    CLN     +01
FAD7 69 00            ADCIM  $00
FAD9 29 07            ANDIM  $07     MAX IS $7FF
FADB 85 07            STA    CLN     +01
FADD 60               RTS

              *LAST LINE DOWN*

FADE 18       LLNDN   CLC
FADF A5 08            LDA    LLN
FAE1 65 14            ADC    CHAPLN
FAE3 85 08            STA    LLN     LLN=LLN+CHAPLN
FAE5 A5 09            LDA    LLN     +01
FAE7 69 00            ADCIM  $00
FAE9 29 07            ANDIM  $07     MAX IS $7FF
FAEB 85 09            STA    LLN     +01
FAED 60               RTS

              *ADJUST THE LINE POINTERS*

FAEE A4 0F    ADJUST  LDY    COL     IS COLUMN NEGATIVE?
FAF0 10 1F            BPL    ADSA    BRANCH ON NO
FAF2 C6 0E            DEC    INLINE  GO BACK ONE LINE
FAF4 10 10            BPL    ADJU    BRANCH IF STILL ON SCREEN
FAF6 20 8E FA         JSR    FLNUP   FIRST LINE UP
FAF9 20 79 FB         JSR    FLNCRT  ADJUST DISPLAY START
FAFC 20 07 FC         JSR    ERAFLN  ERASE THE FIRST LINE
```

```
FAFF 20 AE FA          JSR    LLNUP  LAST LINE UP
FB02 A0 00             LDYIM  $00    RESET INLINE
FB04 84 0E             STY    INLINE

FB06 20 9E FA  ADJU    JSR    CLNUP  CURRENT LINE UP
FB09 A4 14             LDY    CHAPLN
FB0B 88                DEY           COL=CHAPLN-1
FB0C 84 0F             STY    COL
FB0E 4C 35 FB          JMP    ADJCUR


FB11 C4 14     ADSA    CPY    CHAPLN IS COL>=CHAPLN?
FB13 90 20             BCC    ADJCUR BRANCH ON NO
FB15 A0 00             LDYIM  $00
FB17 84 0F             STY    COL    COL=0
FB19 E6 0E             INC    INLINE MOVE DOWN 1 LINE
FB1B A4 0E             LDY    INLINE
FB1D C4 15             CPY    LPSCR  STILL ON SCREEN?
FB1F 90 11             BCC    ADJV   IS INLINE>=LPSCR?
FB21 20 BE FA          JSR    FLNDN  FIRST LINE DOWN
FB24 20 79 FB          JSR    FLNCRT ADJUST DISPLAY START
FB27 20 DE FA          JSR    LLNDN  LAST LINE DOWN
FB2A A4 15             LDY    LPSCR
FB2C 88                DEY
FB2D 84 0E             STY    INLINE INLINE=LPSCR-1
FB2F 20 18 FC          JSR    ERALLN ERASE LAST LINE

FB32 20 CE FA  ADJV    JSR    CLNDN  CURRENT LINE DOWN

FB35 A2 00     ADJCUR  LDXIM  $00
FB37 86 0C             STX    CURSOR RESET CURSOR
FB39 86 0D             STX    CURSOR +01
FB3B A6 0E             LDX    INLINE IS INLINE=0?
FB3D F0 0E             BEQ    ACURX  BRANCH ON YES

FB3F 18        ACURA   CLC
FB40 A5 14             LDA    CHAPLN
FB42 65 0C             ADC    CURSOR
FB44 85 0C             STA    CURSOR
FB46 90 02             BCC    ACURB
FB48 E6 0D             INC    CURSOR +01   CURSOR=INLINE*CHAPLN

FB4A CA        ACURB   DEX
FB4B D0 F2             BNE    ACURA

FB4D 18        ACURX   CLC
FB4E A5 04             LDA    FLN
FB50 65 0C             ADC    CURSOR
FB52 85 0C             STA    CURSOR CURSOR=CURSOR+FLN
FB54 A5 05             LDA    FLN    +01
FB56 65 0D             ADC    CURSOR +01
FB58 85 0D             STA    CURSOR +01

FB5A 18                CLC
FB5B A5 0F             LDA    COL
FB5D 65 0C             ADC    CURSOR CURSOR=CURSOR+COL
```

```
FB5F 85 0C           STA   CURSOR
FB61 90 02           BCC   ACURC
FB63 E6 0D           INC   CURSOR +01

FB65 A2 0E     ACURC LDXIM CURPOH
FB67 A5 0D           LDA   CURSOR +01
FB69 8E 40 C1        STX   AR
FB6C 8D 41 C1        STA   RFILE  CURSOR--->CRT CONTROLLER
FB6F E8              INX
FB70 A5 0C           LDA   CURSOR
FB72 8E 40 C1        STX   AR
FB75 8D 41 C1        STA   RFILE
FB78 60              RTS

               *FIRST LINE TO CRT*

FB79 A2 0C     FLNCRT LDXIM DSPSTH
FB7B A5 05           LDA   FLN    +01
FB7D 8E 40 C1        STX   AR
FB80 8D 41 C1        STA   RFILE
FB83 E8              INX
FB84 A5 04           LDA   FLN
FB86 8E 40 C1        STX   AR
FB89 8D 41 C1        STA   RFILE
FB8C 60              RTS


               *COMPUTE THE CURRENT RAM POINTER*

FB8D 18        CRAMPT CLC
FB8E A5 0A           LDA   SCRPTR
FB90 65 12           ADC   RAMBEG RAMPTR=SCRPTR+RAMBEG
FB92 85 00           STA   RAMPTR
FB94 A5 0B           LDA   SCRPTR +01
FB96 65 13           ADC   RAMBEG +01
FB98 29 D7           ANDIM $D7    MAX IS $D7FF
FB9A 85 01           STA   RAMPTR +01
FB9C 18              CLC
FB9D A5 00           LDA   RAMPTR RAMPTR=RAMBEG+SCRPTR+TEMCOL
FB9F 65 11           ADC   TEMCOL
FBA1 85 00           STA   RAMPTR
FBA3 90 08           BCC   CRAMP
FBA5 E6 01           INC   RAMPTR +01
FBA7 A5 01           LDA   RAMPTR +01
FBA9 29 D7           ANDIM $D7    MAX IS $D7FF
FBAB 85 01           STA   RAMPTR +01

FBAD 60        CRAMP  RTS

               *ERASE TO END OF LINE*

FBAE A6 06     ERTEOL LDX   CLN
FBB0 A4 07           LDY   CLN    +01
FBB2 86 0A           STX   SCRPTR SCRPTR=CLN
FBB4 84 0B           STY   SCRPTR +01
FBB6 A4 0F           LDY   COL    GET THE CURR. COLUMN
```

```
FBB8 84 11              STY    TEMCOL TEMCOL=COL
FBBA A2 20              LDXIM '        SPACE TO X
FBBC A0 00              LDYIM $00     INDEX=0

FBBE 20 8D FB  EREOL    JSR    CRAMPT RAMPTR=RAMBEG+SCRPTR+TEMCOL

FBC1 8A        EROLX    TXA           SPACE--->RAM
FBC2 91 00              STAIY RAMPTR
FBC4 E6 11              INC    TEMCOL TEMCOL=TEMCOL+1
FBC6 E6 00              INC    RAMPTR
FBC8 D0 08              BNE    EROXX
FBCA E6 01              INC    RAMPTR +01 RAMPTR=RAMPTR+1
FBCC A5 01              LDA    RAMPTR +01
FBCE 29 D7              ANDIM $D7     MAX IS $D7FF
FBD0 85 01              STA    RAMPTR +01

FBD2 A5 11     EROXX    LDA    TEMCOL IS TEMCOL>=CHAPLN?
FBD4 C5 14              CMP    CHAPLN
FBD6 90 E9              BCC    EROLX  BRANCH ON NO
FBD8 60                 RTS

               *ERASE TO END OF SCREEN*

FBD9 A6 0E     ERTEOS LDX    INLINE
FBDB E8                 INX           LAST LINE?
FBDC E4 15              CPX    LPSCR
FBDE F0 CE              BEQ    ERTEOL IF YES DO ONLY "EOL"
FBE0 20 AE FB           JSR    ERTEOL DO "EOL" FOR THIS LINE

FBE3 18        EREOS    CLC
FBE4 A5 0A              LDA    SCRPTR
FBE6 65 14              ADC    CHAPLN SCRPTR=SCRPTR+CHAPLN
FBE8 85 0A              STA    SCRPTR OR SCREEN POINTER DOWN
FBEA A5 0B              LDA    SCRPTR +01
FBEC 69 00              ADCIM $00
FBEE 29 07              ANDIM $07     MAX IS $7FF
FBF0 85 0B              STA    SCRPTR +01
FBF2 C5 09              CMP    LLN    +01 IS SCRPTR=LLN?
FBF4 D0 06              BNE    EEOL
FBF6 A5 0A              LDA    SCRPTR
FBF8 C5 08              CMP    LLN
FBFA F0 07              BEQ    EEOLB  ERASE LAST LINE AND STOP

FBFC 84 11     EEOL     STY    TEMCOL TEMCOL=0
FBFE 20 BE FB           JSR    EREOL
FC01 B0 E0              BCS    EREOS


FC03 84 11     EEOLB    STY    TEMCOL TEMCOL=0
FC05 F0 B7              BEQ    EREOL

               *ERASE THE FIRST LINE*

FC07 A0 00     ERAFLN LDYIM $00
FC09 84 11              STY    TEMCOL
FC0B A2 20              LDXIM '
```

```
FC0D A5 04                LDA    FLN
FC0F 85 0A                STA    SCRPTR SCRPTR=FLN
FC11 A5 05                LDA    FLN    +01
FC13 85 0B                STA    SCRPTR +01
FC15 4C BE FB             JMP    EREOL  NOW ERASE THE FIRST LINE

                   *ERASE THE LAST LINE*

FC18 A0 00         ERALLN LDYIM $00
FC1A 84 11                STY    TEMCOL TEMCOL=0
FC1C A2 20                LDXIM '
FC1E A5 08                LDA    LLN
FC20 85 0A                STA    SCRPTR SCRPTR=LLN
FC22 A5 09                LDA    LLN    +01
FC24 85 0B                STA    SCRPTR +01
FC26 4C BE FB             JMP    EREOL  NOW ERASE THE LAST LINE

                   *PUT A CHARACTER ON THE SCREEN*

FC29 18            TVPUT  CLC
FC2A A5 06                LDA    CLN
FC2C 65 12                ADC    RAMBEG
FC2E 85 00                STA    RAMPTR RAMPTR=CLN+RAMBEG
FC30 A5 07                LDA    CLN    +01
FC32 65 13                ADC    RAMBEG +01
FC34 85 01                STA    RAMPTR +01
FC36 A0 00                LDYIM $00
FC38 18                   CLC
FC39 A5 0F                LDA    COL    RAMPTR=CLN+RAMBEG+COL
FC3B 65 00                ADC    RAMPTR
FC3D 85 00                STA    RAMPTR
FC3F 90 08                BCC    TPX
FC41 E6 01                INC    RAMPTR +01
FC43 A5 01                LDA    RAMPTR +01
FC45 29 D7                ANDIM $D7     MAX IS $D7FF
FC47 85 01                STA    RAMPTR +01

FC49 A5 02         TPX    LDA    AHOLD
FC4B 91 00                STAIY RAMPTR
FC4D E6 0F                INC    COL    COL=COL+01
FC4F 4C EE FA             JMP    ADJUST

                   *CURSOR DOWN*

FC52 E6 0E         CURDN  INC    INLINE INLINE=INLINE+1
FC54 20 CE FA             JSR    CLNDN  CURRENT LINE DOWN
FC57 A4 0E                LDY    INLINE
FC59 C4 15                CPY    LPSCR  IS INLINE>=LPSCR?
FC5B 90 11                BCC    CURDNX BRANCH ON NO
FC5D 20 BE FA             JSR    FLNDN  FIRST LINE DOWN
FC60 20 79 FB             JSR    FLNCRT FIRST LINE--->CRT
FC63 20 DE FA             JSR    LLNDN  LAST LINE DOWN
FC66 20 18 FC             JSR    ERALLN ERASE LAST LINE
FC69 A4 15                LDY    LPSCR
FC6B 88                   DEY
FC6C 84 0E                STY    INLINE INLINE=LPSCR-1
```

```
FC6E 4C 35 FB   CURDNX JMP    ADJCUR ADJUST THE CURSOR AND RETURN
                *CURSOR UP*
FC71 C6 0E      CURUP  DEC    INLINE INLINE=INLINE-1
FC73 20 9E FA          JSR    CLNUP  CURRENT LINE UP
FC76 A4 0E             LDY    INLINE IS INLINE=NEGATIVE?
FC78 10 10             BPL    CURUPX BRANCH ON NO
FC7A 20 8E FA          JSR    FLNUP  FIRST LINE UP
FC7D 20 79 FB          JSR    FLNCRT FLN--->CRT
FC80 20 AE FA          JSR    LLNUP  LAST LINE UP
FC83 20 07 FC          JSR    ERAFLN ERASE THE FIRST LINE
FC86 A0 00             LDYIM $00
FC88 84 0E             STY    INLINE INLINE=0
FC8A 4C 35 FB   CURUPX JMP    ADJCUR ADJUST THE CURSOR AND RETURN
                *CURSOR LEFT*
FC8D C6 0F      CURLFT DEC    COL    COL=COL-1
FC8F 4C EE FA          JMP    ADJUST ADJUST THE CURSOR AND RETURN
                *CURSOR RIGHT*
FC92 E6 0F      CURRGT INC    COL    COL=COL+1
FC94 4C EE FA          JMP    ADJUST ADJUST THE CURSOR AND RETURN
                *HOME CURSOR*
FC97 A2 00      HOME   LDXIM $00
FC99 86 0E             STX    INLINE INLINE=0
FC9B 86 0F             STX    COL    COL=0
FC9D A6 04             LDX    FLN
FC9F 86 06             STX    CLN    CLN=FLN
FCA1 A6 05             LDX    FLN    +01
FCA3 86 07             STX    CLN    +01
FCA5 4C EE FA          JMP    ADJUST ADJUST THE CURSOR AND RETURN
                *MASTER RESET OF THE MEMORY MAPPED VDU*
FCA8 A9 00      RESET  LDAIM $00
FCAA 85 04             STA    FLN    FLN=$0000
FCAC 85 05             STA    FLN    +01
FCAE 85 06             STA    CLN    CLN=$0000
FCB0 85 07             STA    CLN    +01
FCB2 85 08             STA    LLN    RESET LAST LINE
FCB4 85 09             STA    LLN    +01
FCB6 85 0F             STA    COL    COL=0
FCB8 85 0E             STA    INLINE INLINE=0
FCBA A6 15             LDX    LPSCR
FCBC CA                DEX           X=LPSCR-1
FCBD 18         RSA    CLC
FCBE A5 14             LDA    CHAPLN
FCC0 65 08             ADC    LLN
```

```
FCC2 85 08             STA    LLN     LLN=(LPSCR-1)*CHAPLN
FCC4 90 02             BCC    RSB
FCC6 E6 09             INC    LLN     +01
FCC8 CA         RSB    DEX
FCC9 D0 F2             BNE    RSA
FCCB 20 D7 FC          JSR    CRTINT SET THE CRT TIMING REGISTERS
FCCE 20 97 FC          JSR    HOME    HOME CURSOR
FCD1 20 D9 FB          JSR    ERTEOS CLEAR THE SCREEN
FCD4 4C 79 FB          JMP    FLNCRT FLN--->CRT AND RETURN
                *INITIALIZE THE CRT CONTROLLER*
FCD7 A2 00      CRTINT LDXIM $00
FCD9 8E 40 C1   CIA    STX    AR      SET THE FILE INDEX
FCDC B5 20             LDAX   TABLE
FCDE 8D 41 C1          STA    RFILE   VALUE--->FILE
FCE1 E8                INX
FCE2 E0 10             CPXIM $10      SET ONLY THE TIMING REGISTERS
FCE4 D0 F3             BNE    CIA
FCE6 60                RTS
                *COMPUTE THE COMMAND ADDRESS INDEX*
FCE7 A2 00      COMCOM LDXIM $00
FCE9 A0 01             LDYIM $01
FCEB BD 09 FD   COMCOA LDAX   COMTAB
FCEE C5 10             CMP    ESCFLG FIND THE COMMAND
FCF0 D0 0A             BNE    COMCOB
FCF2 B9 09 FD          LDAY   COMTAB
FCF5 C5 02             CMP    AHOLD
FCF7 D0 03             BNE    COMCOB
FCF9 18                CLC            C=0 --> X=ADDRESS INDEX
FCFA 90 08             BCC    COMCOC AND RESET THE ESC-FLAG
FCFC E8         COMCOB INX
FCFD E8                INX
FCFE C8                INY
FCFF C8                INY
FD00 E0 36             CPXIM $36
FD02 90 E7             BCC    COMCOA C=1 --> NO COMMAND
FD04 A9 00      COMCOC LDAIM $00
FD06 85 10             STA    ESCFLG RESET THE ESCAPE FLAG
FD08 60                RTS
                *COMMAND TABLE*
FD09 00         COMTAB =      $00
FD0A 0D                =      $0D     <CR>
FD0B 00                =      $00
```

```
FD0C 0A                 =      $0A     <LF>
FD0D 00                 =      $00
FD0E 08                 =      $08     <BS>,<CTL-H> BACK SPACE
FD0F 1B                 =      $1B
FD10 48                 =      'H      <ESC H> CURSOR HOME
FD11 1B                 =      $1B
FD12 41                 =      'A      <ESC A> CURSOR UP
FD13 1B                 =      $1B
FD14 42                 =      'B      <ESC B> CURSOR DOWN
FD15 1B                 =      $1B
FD16 43                 =      'C      <ESC C> CURSOR RIGHT
FD17 1B                 =      $1B
FD18 44                 =      'D      <ESC D> CURSOR LEFT
FD19 1B                 =      $1B
FD1A 4B                 =      'K      <ESC K> ERASE TO EOL
FD1B 1B                 =      $1B
FD1C 4A                 =      'J      <ESC J> ERASE TO EOS

FD1D 00         COTABA  =      $00
FD1E 0B                 =      $0B     <CTL-K> CURSOR UP
FD1F 00                 =      $00
FD20 0C                 =      $0C     <CTL-L> CURSOR RIGHT
FD21 00                 =      $00
FD22 11                 =      $11     <CTL-Q> ERASE TO EOS
FD23 00                 =      $00
FD24 18                 =      $18     <CTL-X> ERASE TO EOL
FD25 00                 =      $00
FD26 1A                 =      $1A     <CTL-Z> CLEAR SCREEN & HOME
FD27 00                 =      $00
FD28 1E                 =      $1E     <CTL-^> CURSOR HOME
FD29 00                 =      $00
FD2A 0A                 =      $0A     <CTL-J> CURSOR DOWN

FD2B 00         COTABB  =      $00
FD2C 10                 =      $10     <CTL-P> SELECT/DESELECT
                                               CENTRONICS
FD2D 1B                 =      $1B
FD2E 52                 =      'R      <ESC><R> DELETE LINE
FD2F 1B                 =      $1B
FD30 2A                 =      '*      <ESC><*> CLEAR SCREEN & HOME
FD31 1B                 =      $1B
FD32 3A                 =      ':      <ESC><:> CLEAR SCREEN & HOME
FD33 1B                 =      $1B
FD34 54                 =      'T      <ESC><T> ERASE TO EOL
FD35 1B                 =      $1B
FD36 74                 =      't      <ESC><t> ERASE TO EOL
FD37 1B                 =      $1B
FD38 59                 =      'Y      <ESC><Y> ERASE TO EOS
FD39 1B                 =      $1B
FD3A 79                 =      'y      <ESC><y> ERASE TO EOS

FD3B 00         COTABX  =      $00
FD3C 06                 =      $06     <CTL-F> SELECT/DESELECT AUTOLF
FD3D 00                 =      $00
FD3E 02                 =      $02     <CTL-B> SEL/DESEL HALF DUPLEX

FD3F FF                 =      $FF
```

```
FD40 FF                  =       $FF
FD41 FF                  =       $FF
FD42 FF                  =       $FF
FD43 FF                  =       $FF
FD44 FF                  =       $FF
FD45 FF                  =       $FF
FD46 FF                  =       $FF
FD47 FF                  =       $FF
FD48 FF                  =       $FF
FD49 FF                  =       $FF
FD4A FF                  =       $FF
FD4B FF                  =       $FF
FD4C FF                  =       $FF
FD4D FF                  =       $FF
FD4E FF                  =       $FF

                *COMMAND ADDRESS TABLE*

FD4F 14         COMADR   =       $14     CR
FD50 FA                  =       $FA
FD51 1E                  =       $1E     LF
FD52 FA                  =       $FA
FD53 2D                  =       $2D     CURSOR LEFT
FD54 FA                  =       $FA
FD55 33                  =       $33     CURSOR HOME
FD56 FA                  =       $FA
FD57 39                  =       $39     CURSOR UP
FD58 FA                  =       $FA
FD59 3F                  =       $3F     CURSOR DOWN
FD5A FA                  =       $FA
FD5B 45                  =       $45     CURSOR RIGHT
FD5C FA                  =       $FA
FD5D 2D                  =       $2D     CURSOR LEFT
FD5E FA                  =       $FA
FD5F 4B                  =       $4B     ERASE TO EOL
FD60 FA                  =       $FA
FD61 51                  =       $51     ERASE TO EOS
FD62 FA                  =       $FA


                *NORMAL VIDEO COMMANDS*

FD63 39                  =       $39     CURSOR UP
FD64 FA                  =       $FA
FD65 45                  =       $45     CURSOR RIGHT
FD66 FA                  =       $FA
FD67 51                  =       $51     ERASE TO EOS
FD68 FA                  =       $FA
FD69 4B                  =       $4B     ERASE TO EOL
FD6A FA                  =       $FA
FD6B 24                  =       $24     CLEAR & HOME
FD6C FA                  =       $FA
FD6D 33                  =       $33     CURSOR HOME
FD6E FA                  =       $FA
FD6F 3F                  =       $3F     CURSOR DOWN
FD70 FA                  =       $FA
```

```
FD71 64                    =      $64    SELECT/DESELECT CENTRONICS
FD72 FA                    =      $FA
FD73 57                    =      $57    DELETE LINE
FD74 FA                    =      $FA
FD75 24                    =      $24    CLEAR & HOME
FD76 FA                    =      $FA
FD77 24                    =      $24      "
FD78 FA                    =      $FA
FD79 4B                    =      $4B    ERASE TO EOL
FD7A FA                    =      $FA
FD7B 4B                    =      $4B     "
FD7C FA                    =      $FA
FD7D 51                    =      $51    ERASE TO EOS
FD7E FA                    =      $FA
FD7F 51                    =      $51       "
FD80 FA                    =      $FA
FD81 6D                    =      $6D    SELECT/DESELECT AUTOLF
FD82 FA                    =      $FA
FD83 76                    =      $76    SEL/DESEL HALF DUPLEX
FD84 FA                    =      $FA
FD85 FF                    =      $FF
FD86 FF                    =      $FF
FD87 FF                    =      $FF
FD88 FF                    =      $FF
FD89 FF                    =      $FF
FD8A FF                    =      $FF
FD8B FF                    =      $FF
FD8C FF                    =      $FF
FD8D FF                    =      $FF
FD8E FF                    =      $FF
FD8F FF                    =      $FF
FD90 FF                    =      $FF
FD91 FF                    =      $FF
FD92 FF                    =      $FF
FD93 FF                    =      $FF
FD94 FF                    =      $FF
                  *CENTRONICS OUTPUT*
FD95 AD 10 C1   CENTRO LDA    VBPBD
FD98 29 08             ANDIM $08    IS THE PRINTER SELECTED?
FD9A F0 29             BEQ    CTROB  PB3 = SEL INPUT
FD9C AD 10 C1          LDA    VBPBD
FD9F 29 10             ANDIM $10     PAPER EMPTY?
FDA1 D0 22             BNE    CTROB  PB4 = PE INPUT
FDA3 A5 02      CTROA  LDA    AHOLD
FDA5 8D 00 C1          STA    VAPBD  OUTPUT THE CHARACTER AND
FDA8 AD 10 C1   WAIT   LDA    VBPBD
FDAB 29 08             ANDIM $08    SEL ?
FDAD F0 16             BEQ    CTROB
FDAF AD 10 C1          LDA    VBPBD
```

```
FDB2 29 10              ANDIM $10     PE ?
FDB4 D0 0F              BNE   CTROB
FDB6 AD 0D C1           LDA   VAIFR   WAIT FOR ACKNOWLEDGE
FDB9 29 10              ANDIM $10     SAMPLE THE CB1 FLAG
FDBB F0 EB              BEQ   WAIT
FDBD AD 0D C1           LDA   VAIFR   RESET CB1 & CB2 FLAGS
FDC0 09 18              ORAIM $18
FDC2 8D 0D C1           STA   VAIFR

FDC5 60         CTROB   RTS



                *INITIALIZE CENTRONICS*

FDC6 A9 FF      INICEN  LDAIM $FF     PB0...PB7 = OUTPUT
FDC8 8D 02 C1           STA   VAPBDD
FDCB A9 A0              LDAIM $A0   CB2 = WRITE HANDSHAKE PULSE OUTPUT
FDCD 8D 0C C1           STA   VAPCR CB1 = NEG. EDGE SENSITIVE
FDD0 AD 12 C1           LDA   VBPBDD
FDD3 29 E7              ANDIM $E7     PB3,PB4 = INPUT
FDD5 8D 12 C1           STA   VBPBDD
FDD8 A9 00              LDAIM $00     RESET THE PRINTER FLAG
FDDA 85 34              STA   CENFLG
FDDC 60                 RTS

                *MOVE THE CRT FILE FROM ROM TO RAM*

FDDD A9 00      MOVCRT  LDAIM $00     REFRESH RAM STARTS AT $D000
FDDF A2 D0              LDXIM $D0
FDE1 85 12              STA   RAMBEG
FDE3 86 13              STX   RAMBEG +01
FDE5 A9 00              LDAIM $00
FDE7 A8                 TAY
FDE8 A6 16              LDX   FORMAT GET THE CURR. FORMAT
FDEA F0 06              BEQ   MCRTB

FDEC 18         MCRTA   CLC
FDED 69 12              ADCIM $12     COMPUTE THE INDEX
FDEF CA                 DEX
FDF0 D0 FA              BNE   MCRTA

FDF2 AA         MCRTB   TAX

FDF3 BD 12 FE   MCRTC   LDAX  CRTINA
FDF6 99 20 00           STAY  TABLE   MOVE TABLE
FDF9 E8                 INX
FDFA C8                 INY
FDFB C0 12              CPYIM $12
FDFD D0 F4              BNE   MCRTC
FDFF 88                 DEY
FE00 B9 20 00           LDAY  TABLE   SET SCREEN PARAMETERS
FE03 85 15              STA   LPSCR
FE05 88                 DEY
FE06 B9 20 00           LDAY  TABLE
FE09 85 14              STA   CHAPLN
```

```
FE0B 60             RTS

                   *INDIRECT NMI & IRQ VECTORS*

FE0C 6C 37 00   NMIJMP JMI    NMIVEC
FE0F 6C 39 00   IRQJMP JMI    IRQVEC


                   *CRT TIMING TABLES*

                   - 80*24 -

FE12 80         CRTINA =      $80  HORIZONTAL TOTAL-1 = 129-1 CHAR.
FE13 50                =      $50  HORIZONTAL DISPLAYED = 80 CHAR.
FE14 60                =      $60  HORIZ. SYNC. POSITION = 96 CHAR.
FE15 08                =      $08  VERT./HORIZ. SYNC WIDTH = 16/8
FE16 22                =      $22  VERTICAL TOTAL-1 = 34 CHAR. LINES
FE17 00                =      $00  VERT. TOTAL ADJ. = 0*64 MICRO SEC.
FE18 18                =      $18  VERTICAL DISPLAYED = 24 LINES
FE19 1C                =      $1C  VERT. SYNC. POSITION= 29 CHAR. LINES
FE1A 00                =      $00  MODE CONTROL
FE1B 08                =      $08  SCAN LINES-1 = 9-1
FE1C 00                =      $00  CURSOR START
FE1D 09                =      $09  CURSOR END
FE1E 00                =      $00  DISPLAY START (NOT NEEDED)
FE1F 00                =      $00
FE20 00                =      $00  CURSOR POSITION (NOT NEEDED)
FE21 00                =      $00
FE22 50                =      $50  CHARACTERS/LINE
FE23 18                =      $18  LINES/SCREEN


                   - 80*25 -

FE24 80                =      $80
FE25 50                =      $50
FE26 60                =      $60
FE27 08                =      $08
FE28 22                =      $22
FE29 00                =      $00
FE2A 19                =      $19
FE2B 1C                =      $1C
FE2C 00                =      $00
FE2D 08                =      $08
FE2E 00                =      $00
FE2F 09                =      $09
FE30 00                =      $00
FE31 00                =      $00
FE32 00                =      $00
FE33 00                =      $00
FE34 50                =      $50
FE35 19                =      $19

                   - 64*16 -

FE36 64                =      $64
```

```
FE37 40              =        $40
FE38 49              =        $49
FE39 05              =        $05
FE3A 16              =        $16
FE3B 0E              =        $0E
FE3C 10              =        $10
FE3D 12              =        $12
FE3E 00              =        $00
FE3F 0C              =        $0C
FE40 00              =        $00
FE41 09              =        $09
FE42 00              =        $00
FE43 00              =        $00
FE44 00              =        $00
FE45 00              =        $00
FE46 40              =        $40
FE47 10              =        $10

                - 64X24 -

FE48 64              =        $64
FE49 40              =        $40
FE4A 52              =        $52
FE4B 05              =        $05
FE4C 22              =        $22
FE4D 00              =        $00
FE4E 18              =        $18
FE4F 1C              =        $1C
FE50 00              =        $00
FE51 08              =        $08
FE52 00              =        $00
FE53 09              =        $09
FE54 00              =        $00
FE55 00              =        $00
FE56 00              =        $00
FE57 00              =        $00
FE58 40              =        $40
FE59 18              =        $18

                - 90X22 -

FE5A 87              =        $87
FE5B 5A              =        $5A
FE5C 6A              =        $6A
FE5D 08              =        $08
FE5E 22              =        $22
FE5F 00              =        $00
FE60 16              =        $16
FE61 1C              =        $1C
FE62 00              =        $00
FE63 08              =        $08
FE64 00              =        $00
FE65 09              =        $09
FE66 00              =        $00
FE67 00              =        $00
FE68 00              =        $00
```

```
FE69 00          =    $00
FE6A 5A          =    $5A
FE6B 16          =    $16

            - 48X12 -

FE6C 46          =    $46
FE6D 30          =    $30
FE6E 3A          =    $3A
FE6F 05          =    $05
FE70 16          =    $16
FE71 0E          =    $0E
FE72 0C          =    $0C
FE73 12          =    $12
FE74 00          =    $00
FE75 0C          =    $0C
FE76 00          =    $00
FE77 09          =    $09
FE78 00          =    $00
FE79 00          =    $00
FE7A 00          =    $00
FE7B 00          =    $00
FE7C 30          =    $30
FE7D 0C          =    $0C

            - 24X24 -

FE7E 38          =    $38
FE7F 18          =    $18
FE80 26          =    $26
FE81 05          =    $05
FE82 22          =    $22
FE83 00          =    $00
FE84 18          =    $18
FE85 1C          =    $1C
FE86 00          =    $00
FE87 08          =    $08
FE88 00          =    $00
FE89 09          =    $09
FE8A 00          =    $00
FE8B 00          =    $00
FE8C 00          =    $00
FE8D 00          =    $00
FE8E 18          =    $18
FE8F 18          =    $18

            - 80X24 (USA) -

FE90 00          =    $00
FE91 00          =    $00
FE92 00          =    $00
FE93 00          =    $00
FE94 00          =    $00
FE95 00          =    $00
FE96 00          =    $00
FE97 00          =    $00
```

```
FE98 00                  =      $00
FE99 00                  =      $00
FE9A 00                  =      $00
FE9B 00                  =      $00
FE9C 00                  =      $00
FE9D 00                  =      $00
FE9E 00                  =      $00
FE9F 00                  =      $00
FEA0 50                  =      $50
FEA1 18                  =      $18

                 *VT 52 DISPLAY*

FEA2 D8          RESVTA CLD
FEA3 78                 SEI
FEA4 A2 FF              LDXIM $FF    RESET THE STACK POINTER
FEA6 9A                 TXS
FEA7 20 63 FF           JSR   INIKBD INIT. PAR. KEYBOARD
FEAA 20 85 FF           JSR   CTLCMD READ ACIA CONTROL/COMMAND REG.
FEAD 20 AE FF           JSR   GETFOT GET THE SCREEN FORMAT
FEB0 20 DD FD           JSR   MOVCRT MOVE THE CRT TABLE & SCREEN FORMAT
FEB3 20 A8 FC           JSR   RESET  INIT. CRT
FEB6 AD 10 C1           LDA   VBPBD  SET READY LOW
FEB9 29 FE              ANDIM $FE
FEBB 8D 10 C1           STA   VBPBD
FEBE AD 12 C1           LDA   VBPBDD
FEC1 09 01              ORAIM $01    PB0=READY-OUTPUT
FEC3 8D 12 C1           STA   VBPBDD
FEC6 A9 00              LDAIM $00
FEC8 85 17              STA   INDEX  NO CHARACTERS IN BUFFER
FECA 85 18              STA   BUFFER
FECC A9 44              LDAIM RECIRQ SET IRQ VECTOR
FECE A2 FF              LDXIM RECIRQ /
FED0 85 39              STA   IRQVEC
FED2 86 3A              STX   IRQVEC +01
FED4 20 C6 FD           JSR   INICEN INITIALIZE CENTRONICS I/O
FED7 A9 00              LDAIM $00
FED9 A2 00              LDXIM $00
FEDB 85 35              STA   DUPLEX SELECT FULL DUPLEX
FEDD 86 36              STX   AUTOLF OUTPUT NO AUTO LF TO CENTRONICS
FEDF 58                 CLI

FEE0 20 76 FF    VTA    JSR   GETKBD
FEE3 D0 2C              BNE   VTAT   ACTIVE KEY?
FEE5 A5 18              LDA   BUFFER
FEE7 F0 F7              BEQ   VTA    ANY CHARACTERS IN BUFFER?
FEE9 A2 00              LDXIM $00

FEEB E8          VTAS   INX
FEEC B5 18              LDAX  BUFFER
FEEE 85 02              STA   AHOLD
FEF0 20 DE F9           JSR   VIDEO
FEF3 A5 34              LDA   CENFLG
FEF5 F0 03              BEQ   VTACEN
FEF7 20 25 FF           JSR   SPECEN OUTPUT THE CHAR. TO THE PRINTER

```

```
FEFA E4 17      VTACEN CPX    INDEX
FEFC D0 ED             BNE    VTAS
FEFE 78                SEI
FEFF A2 00             LDXIM $00
FF01 86 18             STX    BUFFER
FF03 86 17             STX    INDEX
FF05 58                CLI
FF06 AD 10 C1          LDA    VBPBD
FF09 29 FE             ANDIM $FE
FF0B 8D 10 C1          STA    VBPBD  VIA2,PB0 = OUTPUT
FF0E 4C E0 FE          JMP    VTA
FF11 20 D1 FF   VTAT   JSR    ACIOUT
FF14 A5 35             LDA    DUPLEX
FF16 F0 03             BEQ    VTATXY 0=FULL DUPLEX
FF18 20 DE F9          JSR    VIDEO  ECHO TO SCREEN
FF1B A5 34      VTATXY LDA    CENFLG
FF1D F0 C1             BEQ    VTA    OUTPUT TO CENTRONICS?
FF1F 20 25 FF          JSR    SPECEN
FF22 4C E0 FE          JMP    VTA
                *SPECIAL CENTRONICS OUTPUT*
FF25 A5 36      SPECEN LDA    AUTOLF
FF27 F0 17             BEQ    VTATZX ADD AN AUTOLF?
FF29 A5 02             LDA    AHOLD
FF2B C9 0D             CMPIM $0D     CHECK FOR CR
FF2D D0 11             BNE    VTATZX
FF2F 20 95 FD          JSR    CENTRO OUTPUT A CR
FF32 A5 02             LDA    AHOLD
FF34 48                PHA
FF35 A9 0A             LDAIM $0A
FF37 85 02             STA    AHOLD
FF39 20 95 FD          JSR    CENTRO OUTPUT A LF
FF3C 68                PLA
FF3D 85 02             STA    AHOLD
FF3F 60                RTS
FF40 20 95 FD   VTATZX JSR    CENTRO OUTPUT THE CURR. CHAR.
FF43 60                RTS
FF44 48         RECIRQ PHA
FF45 AD 10 C1          LDA    VBPBD
FF48 09 01             ORAIM $01     READY=HIGH
FF4A 8D 10 C1          STA    VBPBD
FF4D 8A                TXA
FF4E 48                PHA
FF4F AD 31 C1          LDA    ACIASR RESET THE IRQ LINE
FF52 E6 17             INC    INDEX
FF54 A6 17             LDX    INDEX
```

```
FF56 AD 30 C1                  LDA   RECREG GET THE CHARACTER
FF59 95 18                     STAX  BUFFER AND STORE IT IN THE FIFO
FF5B A9 01                     LDAIM $01
FF5D 85 18                     STA   BUFFER THERE ARE CHAR. IN THE FIFO
FF5F 68                        PLA
FF60 AA                        TAX
FF61 68                        PLA
FF62 40                        RTI




                        *INITIALIZE KEYBOARD*

FF63 A9 01              INIKBD LDAIM $01    ENABLE INPUT LATCHING
FF65 8D 0B C1                  STA   VAACR
FF68 AD 0C C1                  LDA   VAPCR
FF6B 09 01                     ORAIM $01    CA1 = ACTIVE POSITIVE EDGE
FF6D 8D 0C C1                  STA   VAPCR
FF70 A9 00                     LDAIM $00    PA0...PA7=INPUT
FF72 8D 03 C1                  STA   VAPADD
FF75 60                        RTS


                        *GET KEYBOARD*

FF76 AD 0D C1           GETKBD LDA   VAIFR
FF79 29 02                     ANDIM $02    CA1 FLAG = 1?
FF7B F0 07                     BEQ   GETKA
FF7D AD 01 C1                  LDA   VAPAD  READ KEYBOARD DATA AND
FF80 29 7F                     ANDIM $7F
FF82 85 02                     STA   AHOLD    RESET CA1 FLAG

FF84 60                 GETKA  RTS


                        *GET CONTROL & COMMAND REGISTER OF THE ACIA*


FF85 A9 02              CTLCMD LDAIM $02    VBPB1 = OUTPUT
FF87 8D 12 C1                  STA   VBPBDD
FF8A A2 00                     LDXIM $00    VBPB1 = LOW
FF8C 8E 10 C1                  STX   VBPBD
FF8F AD 11 C1                  LDA   VBPAD  GET ACICMD
FF92 49 FF                     EORIM $FF
FF94 8D 32 C1                  STA   ACICMD
FF97 A9 04                     LDAIM $04    VBPB2 = OUTPUT
FF99 8D 12 C1                  STA   VBPBDD
FF9C 8E 10 C1                  STX   VBPBD  VBPB2 = LOW
FF9F AD 11 C1                  LDA   VBPAD  GET ACICTL
FFA2 49 FF                     EORIM $FF
FFA4 8D 33 C1                  STA   ACICTL
FFA7 8E 13 C1                  STX   VBPADD VBPAD = INPUT
FFAA 8E 12 C1                  STX   VBPBDD VBPBD = INPUT
FFAD 60                        RTS
```

```
                        *GET FORMAT*

FFAE AD 12 C1   GETFOT  LDA    VBPBDD
FFB1 29 1F              ANDIM  $1F      PB7...PB5 = INPUT
FFB3 8D 12 C1           STA    VBPBDD
FFB6 AD 10 C1           LDA    VBPBD
FFB9 49 FF              EORIM  $FF
FFBB A2 05              LDXIM  $05

FFBD 4A         GETFA   LSRA
FFBE CA                 DEX
FFBF D0 FC              BNE    GETFA
FFC1 85 16              STA    FORMAT
FFC3 60                 RTS

                        *ACIA RECEIVER*

FFC4 A9 08      ACIIN   LDAIM  $08      RECEIVER MASK
FFC6 2D 31 C1           AND    ACIASR   RECEIVER FULL?
FFC9 F0 F9              BEQ    ACIIN
FFCB AD 30 C1           LDA    RECREG
FFCE 85 02              STA    AHOLD    READ & SAVE THE CHAR.
FFD0 60                 RTS

                        ACIA TRANSMITTER*

FFD1 A9 10      ACIOUT  LDAIM  $10      TRANSMITTER MASK
FFD3 2D 31 C1           AND    ACIASR   TRANSMITTER EMPTY?
FFD6 F0 F9              BEQ    ACIOUT
FFD8 A5 02              LDA    AHOLD
FFDA 8D 30 C1           STA    TRAREG   GET AND TRANSMIT THE CHAR.
FFDD 60                 RTS

                        *CHECK TERMINAL ROUTINE*

FFDE D8         CHE     CLD
FFDF 78                 SEI
FFE0 A2 FF              LDXIM  $FF
FFE2 9A                 TXS
FFE3 20 63 FF           JSR    INIKBD
FFE6 20 AE FF           JSR    GETFOT
FFE9 20 DD FD           JSR    MOVCRT
FFEC 20 A8 FC           JSR    RESET

FFEF 20 76 FF   CHECKA  JSR    GETKBD
FFF2 F0 FB              BEQ    CHECKA
FFF4 20 DE F9           JSR    VIDEO
FFF7 4C EF FF           JMP    CHECKA

-T
```

```
SYMBOL TABLE 3400 386E
ACIASR C131    ACICMD C132    ACICTL C133    ACIIN  FFC4
ACIOUT FFD1    ACURA  FB3F    ACURB  FB4A    ACURC  FB65
ACURX  FB4D    ADJCUR FB35    ADJU   FB06    ADJUST FAEE
ADJV   FB32    ADSA   FB11    AHOLD  0002    AR     C140
AUTOLF 0036    BRKTST F9F6    BUFFER 0018    CENFLG 0034
CENTRO FD95    CHAPLN 0014    CHECK  F9FE    CHECKA FFEF
CHE    FFDE    CIA    FCD9    CLNDN  FACE    CLNUP  FA9E
CLN    0006    CLRHOM FA24    COL    000F    COMADR FD4F
COMCOA FCEB    COMCOB FCFC    COMCOC FD04    COMCOM FCE7
COMTAB FD09    COTABA FD1D    COTABB FD2B    COTABX FD3B
CRAMP  FBAD    CRAMPT FB8D    CRTINA FE12    CRTINT FCD7
CTLCMD FF85    CTROA  FDA3    CTROB  FDC5    CURDN  FC52
CURDNX FC6E    CUREND 000B    CURLFT FC8D    CURPOH 000E
CURPOL 000F    CURRGT FC92    CURSOR 000C    CURSTA 000A
CURUP  FC71    CURUPX FC8A    DELLIN FA57    DOWN   FA3F
DSPSTH 000C    DSPSTL 000D    DUMMY  001F    DUPLEX 0035
EEOL   FBFC    EEOLB  FC03    ERAFLN FC07    ERALLN FC18
EREOL  FBBE    EREOS  FBE3    ERLNX  FA4B    EROLX  FBC1
EROXX  FBD2    ERSCRX FA51    ERTEOL FBAE    ERTEOS FBD9
ESCFLG 0010    ESC    001B    FEED   FA1E    FLNCRT FB79
FLNDN  FABE    FLNUP  FA8E    FLN    0004    FORMAT 0016
GETFA  FFBD    GETFOT FFAE    GETKA  FF84    GETKBD FF76
HOCU   FA33    HOME   FC97    HORDIS 0001    HORTOT 0000
HSYPOS 0002    INDEX  0017    INICEN FDC6    INIKBD FF63
INLINE 000E    IRQJMP FE0F    IRQVEC 0039    JMPVEC 0032
LEFT   FA2D    LIPENH 0010    LIPENL 0011    LLNDN  FADE
LLNUP  FAAE    LLN    0008    LPSCR  0015    MCRTA  FDEC
MCRTB  FDF2    MCRTC  FDF3    MODE   0008    MOVCRT FDDD
NMIJMP FE0C    NMIVEC 0037    RAMBEG 0012    RAMPTR 0000
RECIRQ FF44    RECREG C130    RESET  FCA8    RESVTA FEA2
RETURN FA14    RFILE  C141    RIGHT  FA45    RSA    FCBD
RSB    FCC8    SCANLN 0009    SCRPTR 000A    SPACE  0020
SPECEN FF25    TABLE  0020    TEMCOL 0011    TOGDUP FA76
TOGGLE FA64    TOGLF  FA6D    TOSCR  FA88    TPX    FC49
TRAREG C130    TVPUT  FC29    UP     FA39    UPDATH 0012
UPDATL 0013    VAACR  C10B    VAIER  C10E    VAIFR  C10D
VALID  FA7F    VALVEC FA11    VAPAD  C101    VAPADD C103
VAPADN C10F    VAPBD  C100    VAPBDD C102    VAPCR  C10C
VASR   C10A    VATACH C105    VATACL C104    VATALH C107
VATALL C106    VATBCH C109    VATBCL C108    VBACR  C11B
VBIER  C11E    VBIFR  C11D    VBPAD  C111    VBPADD C113
VBPADN C11F    VBPBD  C110    VBPBDD C112    VBPCR  C11C
VBSR   C11A    VBTACH C115    VBTACL C114    VBTALH C117
VBTALL C116    VBTBCH C119    VBTBCL C118    VERDIS 0006
VERTOT 0004    VHSYWI 0003    VIDEND F9EB    VIDEO  F9DE
VSYPOS 0007    VTACEN FEFA    VTAS   FEEB    VTAT   FF11
VTATXY FF1B    VTATZX FF40    VTA    FEE0    VTOTAJ 0005
WAIT   FDA8
-T1
```

```
SYMBOL TABLE 3400 386E
HORTOT 0000    RAMPTR 0000    HORDIS 0001    AHOLD  0002
HSYPOS 0002    VHSYWI 0003    FLN    0004    VERTOT 0004
VTOTAJ 0005    CLN    0006    VERDIS 0006    VSYPOS 0007
LLN    0008    MODE   0008    SCANLN 0009    CURSTA 000A
SCRPTR 000A    CUREND 000B    CURSOR 000C    DSPSTH 000C
DSPSTL 000D    CURPOH 000E    INLINE 000E    COL    000F
CURPOL 000F    ESCFLG 0010    LIPENH 0010    LIPENL 0011
TEMCOL 0011    RAMBEG 0012    UPDATH 0012    UPDATL 0013
CHAPLN 0014    LPSCR  0015    FORMAT 0016    INDEX  0017
BUFFER 0018    ESC    001B    DUMMY  001F    SPACE  0020
TABLE  0020    JMPVEC 0032    CENFLG 0034    DUPLEX 0035
AUTOLF 0036    NMIVEC 0037    IRQVEC 0039    VAPBD  C100
VAPAD  C101    VAPBDD C102    VAPADD C103    VATACL C104
VATACH C105    VATALL C106    VATALH C107    VATBCL C108
VATBCH C109    VASR   C10A    VAACR  C10B    VAPCR  C10C
VAIFR  C10D    VAIER  C10E    VAPADN C10F    VBPBD  C110
VBPAD  C111    VBPBDD C112    VBPADD C113    VBTACL C114
VBTACH C115    VBTALL C116    VBTALH C117    VBTBCL C118
VBTBCH C119    VBSR   C11A    VBACR  C11B    VBPCR  C11C
VBIFR  C11D    VBIER  C11E    VBPADN C11F    RECREG C130
TRAREG C130    ACIASR C131    ACICMD C132    ACICTL C133
AR     C140    RFILE  C141    VIDEO  F9DE    VIDEND F9EB
BRKTST F9F6    CHECK  F9FE    VALVEC FA11    RETURN FA14
FEED   FA1E    CLRHOM FA24    LEFT   FA2D    HOCU   FA33
UP     FA39    DOWN   FA3F    RIGHT  FA45    ERLNX  FA4B
ERSCRX FA51    DELLIN FA57    TOGGLE FA64    TOGLF  FA6D
TOGDUP FA76    VALID  FA7F    TOSCR  FA88    FLNUP  FA8E
CLNUP  FA9E    LLNUP  FAAE    FLNDN  FABE    CLNDN  FACE
LLNDN  FADE    ADJUST FAEE    ADJU   FB06    ADSA   FB11
ADJV   FB32    ADJCUR FB35    ACURA  FB3F    ACURB  FB4A
ACURX  FB4D    ACURC  FB65    FLNCRT FB79    CRAMPT FB8D
CRAMP  FBAD    ERTEOL FBAE    EREOL  FBBE    EROLX  FBC1
EROXX  FBD2    ERTEOS FBD9    EREOS  FBE3    EEOL   FBFC
EEOLB  FC03    ERAFLN FC07    ERALLN FC18    TVPUT  FC29
TPX    FC49    CURDN  FC52    CURDNX FC6E    CURUP  FC71
CURUPX FC8A    CURLFT FC8D    CURRGT FC92    HOME   FC97
RESET  FCA8    RSA    FCBD    RSB    FCC8    CRTINT FCD7
CIA    FCD9    COMCOM FCE7    COMCOA FCEB    COMCOB FCFC
COMCOC FD04    COMTAB FD09    COTABA FD1D    COTABB FD2B
COTABX FD3B    COMADR FD4F    CENTRO FD95    CTROA  FDA3
WAIT   FDA8    CTROB  FDC5    INICEN FDC6    MOVCRT FDDD
MCRTA  FDEC    MCRTB  FDF2    MCRTC  FDF3    NMIJMP FE0C
IRQJMP FE0F    CRTINA FE12    RESVTA FEA2    VTA    FEE0
VTAS   FEEB    VTACEN FEFA    VTAT   FF11    VTATXY FF1B
SPECEN FF25    VTATZX FF40    RECIRQ FF44    INIKBD FF63
GETKBD FF76    GETKA  FF84    CTLCMD FF85    GETFOT FFAE
GETFA  FFBD    ACIIN  FFC4    ACIOUT FFD1    CHE    FFDE
CHECKA FFEF
```